Сканеры работают в Linux! с. 14

LINUX FORMAT

Главное в мире Linux

Май 2010 № 5 (131)

ВАШ БЕСПЛАТНЫЙ DVD

MOPSLinux 7.0

Лучше, чем Slackware

А также: LFS 6.6 » KDE 4.4 » OOo 3.2 » Ответы

Linux изнутри

Узнайте, как работает ваш компьютер, и попробуйте сами! с. 22

Grub » D-Bus » SSH » Ядро
Брандмауэр » PAM и так далее

32 страницы учебников на любой вкус!

Cygwin

Наш ответ Windows Services for Unix — на с. 32

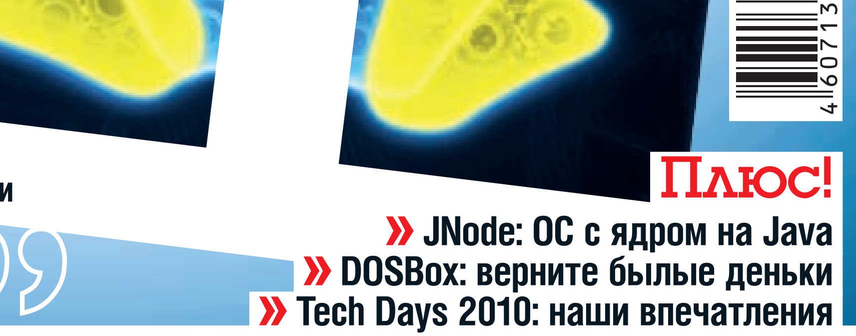
ClamAV за минуту

Настройте свободный антивирус и защитите Linux и Windows с. 34

Время Nano!

Попробуйте текстовый редактор, не встревающий в спор Vi и Emacs с. 38





«Легко представить, как наши аппараты помогают при спасательных работах»

Николас Рой, вертолетостроитель с. 30

Плюс!

» JNode: ОС с ядром на Java
» DOSBox: верните былые деньки
» Tech Days 2010: наши впечатления

» Boxee
Дни MythTV сочтены? Насколько это реально, узнайте сами на с. 60

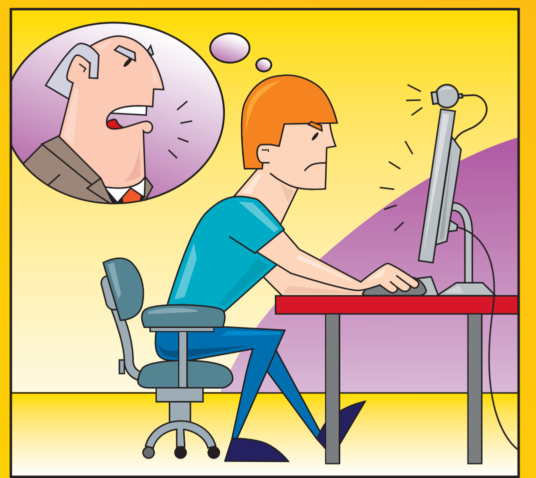
» TaskJuggler
Управляйте проектами с помощью свободного ПО с. 72

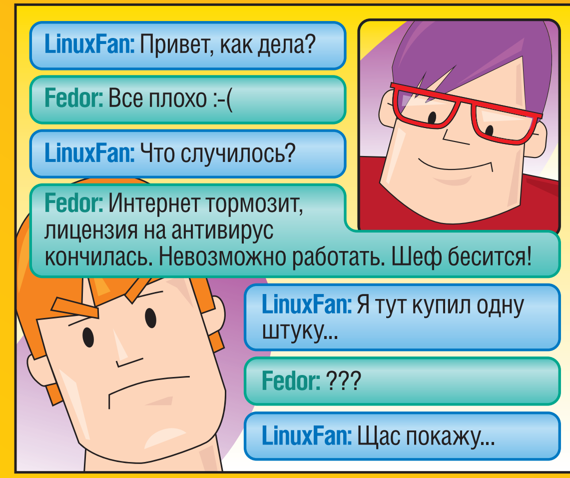
» Python
Расширьте свои навыки в программировании графики с. 76

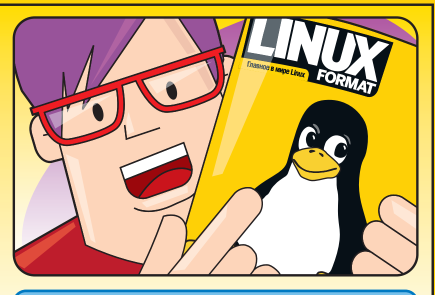
Подписной индекс в каталоге агентства «Роспечать» 20882
Подписной индекс в каталоге «Пресса России» 87974
Подписной индекс в каталоге «Почта России» 16572











LinuxFan: Вот это тема! В этом журнале – все о Linux!

LinuxFan: И в каждом номере диск с кучей свободного ПО!

Fedor: Каждый раз идти в магазин, тратить кучу денег...

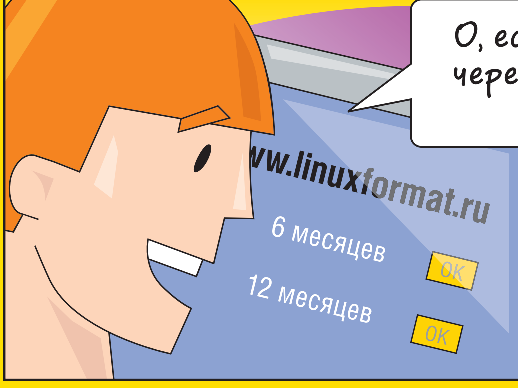


LinuxFan: Зачем? Подпишись на журнал и сэкономь 40 %!

Fedor: Серьезно? Вот это да!

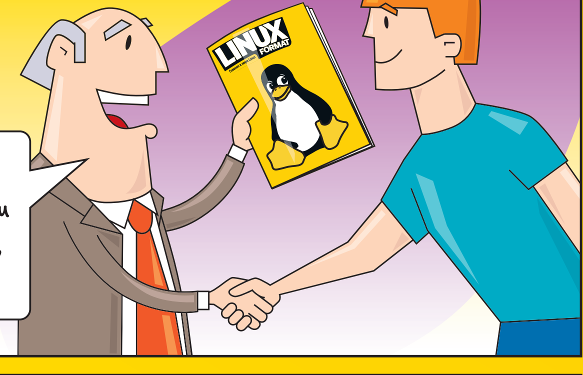


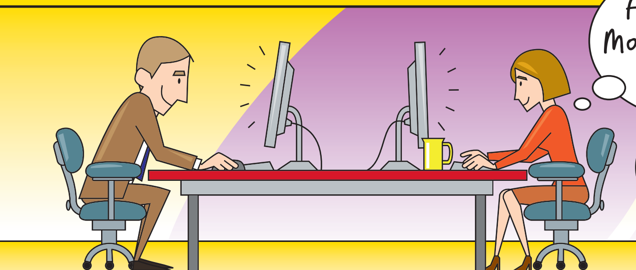
Еще и экономия в 40 %? Надо брать!



О, есть подписка через интернет! Класс!

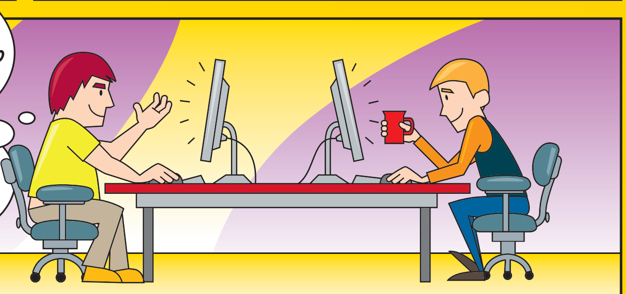
Спасибо, Федор! С приходом Linux в наш офис работа наконец наладилась!





Наконец-то! Можно спокойно работать!

И ничего не глючит!







ПОДПИСНЫЕ ИНДЕКСЫ В КАТАЛОГАХ:	Агентство «Роспечать»	20882
	«Пресса России»	87974
	«Почта России»	16572

www.linuxformat.ru



Что мы делаем

» Мы поддерживаем открытое сообщество, предоставляя источник информации и площадку для обмена мнениями.

» Мы помогаем всем читателям получить от Linux максимум пользы, публикуя статьи в разделе «Учебники»: здесь каждый найдет что-то по своему вкусу.

» Мы выпускаем весь код, появляющийся на страницах раздела «Учебники», по лицензии GNU GPLv3.

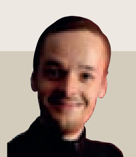
» Мы стремимся предоставлять точные, актуальные и непредвзятые сведения обо всем, что касается Linux и свободного ПО.

Кто мы

Тема этого номера – внутреннее устройство Linux. Поэтому мы спросили у наших экспертов: а какой секретный ингредиент делает Linux таким замечательным?



Грэм Моррисон
Сотни маленьких пингвинчиков, крутящихся в колесах, дают больше энергии. И симпотности.



Майк Сондерс
В **mm/slab.c** в исходниках ядра содержится полная копия ДНК бороды Столлмена. Вот так-то.



Нейл Ботвик
Это, должно быть, тот самый код, что был украден у SCO. Ведь правда, Дарл?



Эфраин Эрнандес-Мендоса
Сообщения об ошибках, действительно привлекающие внимание, скажем: «Ваш принтер сгорел».



Эндрю Грегори
Вустерский соус – секретный ингредиент всего, что я делаю. На чизкейк не желаете?



Энди Ченнел
Наверное, в именах функций вроде **drain_alien_cache()** в исходниках ядра есть что-то магическое.



Дэвид Картрайт
Ключ к успеху ПО – агрессивная оптимизация. Из всех двоичных разрядов Linux использует 1.



Энди Хадсон
Чтобы узнать этот секрет, нажмите все клавиши разом, ровно в 13:37, в момент затмения.



Ник Вейч
Магия? Кто в нее верит? Пардон, кажется, я не выключил котел. Где же глаз летучей мыши...



Сюзан Линтон
Если переименовать vmlinuz в zunilmv, загрузочные сообщения будут прокручиваться снизу вверх.



Шашанк Шарма
Мы отклонились от темы. Секретный ингредиент, конечно, любовь. Любовь и 12,6 млн. строк кода.



Маянк Шарма
Хорошо, я вам скажу. Секретный ингредиент, который делает Linux таким... ой, кто-то звонит в дверь!



Что внутри?

Давным-давно, когда массовые компьютеры еще только появились в нашей стране, а свободное ПО (как ни странно) уже делало первые робкие попытки поселиться на них, был выдвинут тезис: быть просто пользователем Linux – неинтересно. В те суровые времена это, пожалуй, было верно лишь наполовину: Linux середины девяностых едва ли можно было пользоваться, не вникая в то, как работает система, было ли вам это интересно или нет. Видимо, в те годы за пользователями свободной ОС и закрепилась слава «бородатых ребят в сандалиях», денно и нощно сидящих в командной строке.

За прошедшие годы многое изменилось. Мы не устаем повторять, что Linux стал дружелюбнее и едва ли вызовет затруднения у человека, мало-мальски знакомого с компьютером. Подросшее сообщество его пользователей, вышедшее далеко за пределы технических кругов – лишнее тому подтверждение. Но сегодня важно не это – важно то, что внутри себя Linux во многом остается таким же, каким был десять-пятнадцать лет назад.

Я уже чувствую, как вы готовы возмутиться: а как же *HAL*, *NetworkManager*, мириады других средств, призванных превратить черный экран в удобное графическое окружение? Никто не собирается умалять их достоинств – но они являются лишь надстройками над фундаментом системы, причем – надстройками необязательными.

Однако упомянутое многообразие создает густой лес, в котором необходимо ориентироваться. Увы, мы не можем «прокачать» каждого до уровня Linux-гуру за один номер журнала, но в этом выпуске вы найдете все необходимое, чтобы углубить свои знания об устройстве свободной ОС. Читайте материал на стр. 22, слушайте лекции, устанавливайте LFS – в общем, наслаждайтесь!

Валентин Синицын, главный редактор
» info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Вопросы распространения: sales@linuxformat.ru
Web-сайт: www.linuxformat.ru
› Адрес редакции: Россия, Санкт-Петербург, Лиговский пр., 50, корп. 15
› Телефон редакции: (812) 309-06-86. Дополнительная информация на с. 112



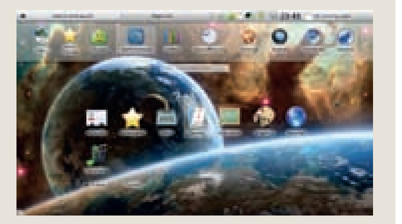
Содержание

Весь номер – прямо как на ладони: приятного чтения!

Обзоры



KDE 4.4 10
После ожиданий, разочарований и смятений, дадим ли мы KDE 4 еще один шанс завоевать наши симпатии?



› И вновь KDE 4 стремится завлечь нас красивой картинкой.

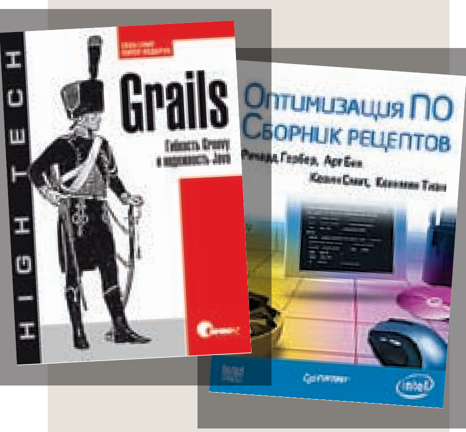
OpenOffice.org 3.2 11
Наконец-то это свершилось: флагманский офисный пакет свободного ПО стал быстрее. Заметно быстрее.

PogoPlug 12
Создать собственное вычислительное облако? С моей-то репутацией? Да о чем они вообще там думают!

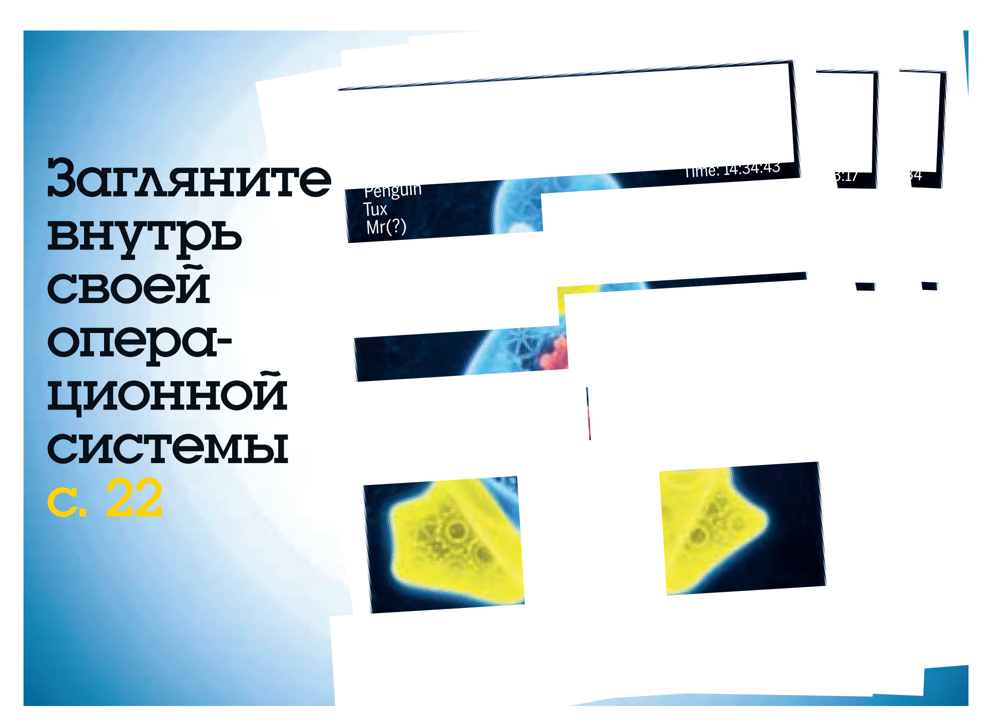
Denix 2.9 13
Угадайте, кто написал к этому дистрибутиву на основе Ubuntu руководство для начинающих пользователей?

VueScan 8 14
Сканеры поддерживаются в Linux с переменным успехом, и мы рукоплещем программе, способной с ними сговориться.

Книжное обозрение 15



› Создавайте web-приложения или пишите оптимальный код на C/C++.



Загляните внутрь своей операционной системы с. 22

Что за штука...

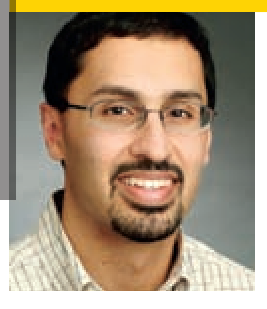
MeeGo
Скоро — на мобильных устройствах в ближайшем к вам магазине с. 48



Nano с. 38
Забудьте *Emacs* и *Vi* — вот единственный настоящий текстовый редактор.



Люди говорят



“Куда-то не добраться вертолетам, а куда-то — наземным роботам.”

Николас Рой, отец Skynet 2.0 с. 30

MOPSLinux 7.0 и LFS 6.6

Системы для желающих понять, как работает Linux, плюс

» **Лекции**, читаемые на факультете ВМиК МГУ
» **KDE 4.4 Live** Попробуйте новый рабочий стол без установки на диск
» **OpenOffice.org 3.2** Свободный офисный пакет
...и множество других приложений и игр! **с. 100**

LXFDVD131

Ищите в этом номере...

Делитесь лучшим! 32
Добавьте на рабочий стол Windows немного свободного ПО при помощи *Cygwin*.

Боритесь с вирусами! 34
ClamAV отловит вредоносный код, поступающий на ваш компьютер, верите вы в него или нет.

Ядро на Java 42
Вопреки убеждению, «кофейные» языки тоже подходят для системного программирования – узнайте, почему.

Go, часть вторая 44
Завершите свое знакомство с новейшим языком из лабораторий Google с элементами ООП.

Постоянные рубрики

Новости 4
Ура, конференции!

Интервью LXF 30
Николас Рой, MIT.

Что за штука 48
MeeGo – мобильная Linux-платформа, созданная на основе Intel и Maemo.

Рубрика сисадмина 50
Доктор посылает правильные сигналы, разбирается с RPM и ищет лекарство от рака.

Ответы 88
Проблемы Linux решены: от 3G до PS3.

Hotpicks 94
Лучшие в мире новинки свободного ПО.

Диск Linux Format 100
Все необходимое для желающих разобраться в работе Linux.

Пропустили номер? ... 107
Закажите один из предыдущих выпусков журнала.

Школа LXF 108
Свободное ПО на уроке астрономии.

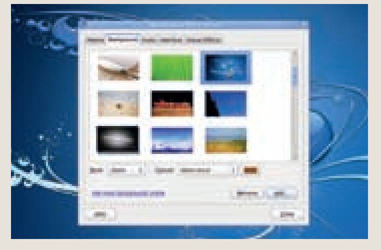
Через месяц 112
LXF132: не испортите себе сюрприз!



› А вы не забыли оформить подписку на LXF?

Учебники

Начинающим
Рабочий стол 56
Сделайте свое окружение красивым, а также безопасным для детей.



› Искусство принимать решения — это умение говорить «нет» вместо «да». Да.

Boxee
Больше телевидения 60
То есть больше того, что вы хотите, и когда вы хотите, в этом потенциальном убийце *MythTV*.

DOSBox
Играем в ретро-игры 64
Дайте волю своей ностальгии – в старые добрые времена игры действительно были лучше.

R
Момент истины 68
Свободный статистический инструментарий вскрывает единство российского студенчества с Орденом иезуитов.

TaskJuggler
Управляем проектами 72
Работайте, чтобы жить, а не живите, чтобы работать – *TaskJuggler* подскажет, что и когда для этого надо делать.

Python
Кодируем с Cairo 76
В этот раз мы призовем на помощь стороннюю библиотеку и создадим симпатичные часики (о вкусах не спорят).

PING и BackupPC
Резервирование 80
Сохраните наработанное – будь то тщательно настроенная ОС или просто документы – в надежном месте.

Hardcore Linux
Программный RAID 84
Хороший способ повысить производительность и надежность, не используя специального оборудования.

ГЛАВНОЕ HTML5 vs Adobe Flash » Особенности национального «энтерпрайза» » Госконтракты » Linux Foundation расширяется » Что было?

Битва титанов

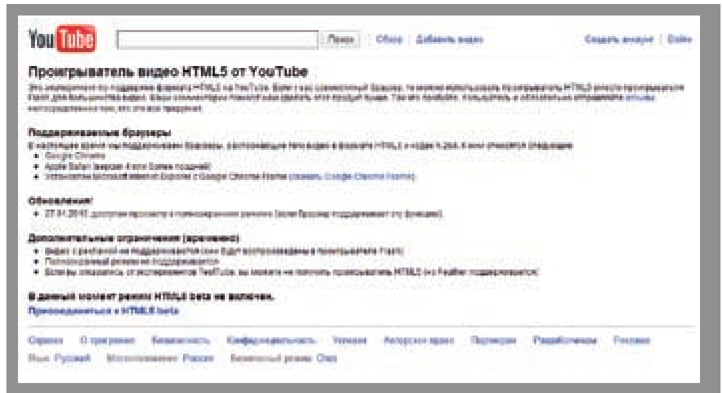


» Рубрику ведет **ЕВГЕНИЙ КРЕСТНИКОВ**

Споры вокруг Adobe Flash и технологий HTML5 неожиданно возобновились. Поводом для очередного этапа баталий стали недавние заявления главы Apple.

Стив Джобс [Steve Jobs] обвинил разработчиков Flash в лени и нежелании оптимизировать технологию для продуктов Apple. По его мнению, использование Flash приводит к быстрому разряду аккумуляторов мобильных устройств, и потому эта технология не поддерживается в iPhone OS. От слов в Apple тут же перешли к делу – лицензионное соглашение iPhone OS 4.0 SDK содержит пункты, запрещающие использовать сторонние средства разработки, в том числе компиляторы Flash для iPhone. Похоже, скоро дело дойдет до судебных слушаний – по данным портала IT World, Adobe собирается подать иск против Apple.

Сторонникам свободного ПО корпоративные скандалы не интересны. Главное здесь, что отказ от Flash в пользу свободных стандартов получил поддержку с весьма неожиданной стороны: корпорация Apple даже выпустила список ресурсов, готовых к работе с iPad. Все они перешли на трансляцию потокового видео с использованием HTML5, либо имеют специальную версию, не требующую поддержки Flash. Спор в очередной раз скатился к обсуждению способов видеотрансляции в Сети. Впрочем, SVG и Canvas вполне способны заменить технологию Adobe для создания web-интерфейсов, игр и анимации (хотя они тоже могут прилично загрузить процессор). Сейчас мнения компаний разделились: Apple и Microsoft планируют поддерживать в браузерах стандарт H.264. Мобильные устройства Apple имеют аппаратный декодер H.264, и выбор формата здесь очевиден (Nokia отказывается поддерживать Theora в своей мобильной платформе по тем же соображениям). С Microsoft все сложнее – сейчас все ждут официального анонса *IE 9* (несмотря на падение популярности, браузером Microsoft пользуется большинство обитателей Сети), и пока не ясно, какие форматы он будет использовать. Google реализовал в *Chrome* поддержку обоих кодеков, но на портале YouTube (кстати, сейчас проходит тестирование его версия, созданная с использованием тэга **<video>**) предпочитает применять H.264. Наиболее принципиальную позицию заняла Mozilla: компания наотрез отказывается включать реализацию H.264 в свои продукты.



» В принципе, YouTube уже можно смотреть на компьютерах без установленного Flash.

Представители Opera Software более осторожны в высказываниях. Директор по стандартам Чарльз МакКетиНевил [Charles McCathieNevile], к которому мы обратились за комментариями, сказал, что они не в восторге от Flash, но поскольку есть инструменты и люди, которые используют их для разработки, говорить об отказе от этой технологии еще рано. Что касается выбора кодека, *Opera* сейчас поддерживает Theora. Однако г-н МакКетиНевил заявил, что при необходимости будет реализована поддержка других форматов: DIRAC, VP8 или H.264. Кроме того, формат SVG реализован в настольной версии *Opera* наиболее полно. Говоря о качестве кодеков, он нашел их сравнимыми: «Я не эксперт в области кодирования видео, но считаю, что они на самом деле похожи по уровню качества. Таким образом, разница сводится к лицензированию. Насколько я понимаю, лучшие реализации H.264 сегодня ненамного лучше Theora». Что касается возможности включения H.264 в стандарт HTML5, спикер Opera счел ее маловероятной: «W3C стремится к стандартам, не требующим лицензионных отчислений, и я не думаю, что H.264 будет исключением», сказал он.

Позиция Adobe еще интереснее. У нас сложилось стойкое ощущение, что в понимании компании никакого HTML5 пока нет, и появится он не скоро. Конечно, до принятия стандарта еще далеко, однако отдельные его элементы поддерживаются браузерами уже сейчас (либо вот-вот начнут поддерживаться), и отмахиваться от этого факта мы бы не стали. В любом случае, на фоне этого конфликта назревает очередной передел рынка браузеров, и кто выйдет победителем, пока сказать сложно. Разработчики устройств и коммерческих программ предпочтут лицензировать H.264, в то время, как сторонники открытого (или просто бесплатного) ПО склоняются к использованию Theora.

К слову

Корпорация Google выделила средства на проект TheorARM, с целью создания оптимизированной для процессоров ARM реализации кодека Theora, которая позволит воспроизводить HD-видео в этом формате на маломощных устройствах. Кроме того, в мае корпорация планирует открыть принадлежащий ей формат VP8. Есть мнение, что это позволит достичь компромисса между качеством и лицензионными ограничениями.

Несите рублики!

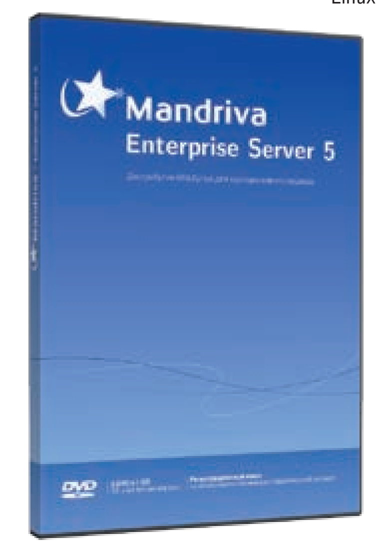
«Продается не ПО, а сервис» – знакомый тезис? Попробуем разобраться, кто готов следовать ему в нашей стране.

Так уж сложилось, что в новостной рубрике мы освещаем в основном некоммерческие реализации Linux. Сегодня же мы решили посмотреть, как обстоят дела у компаний, предлагающих для своих дистрибутивов платную поддержку. Причем нас пока не интересуют государственные программы (например, школьный Linux) – мы поговорим о них ниже. Согласитесь, случай, когда заказчик оплачивает сервис из собственного кармана, гораздо интересней. В качестве объекта исследований возьмем апрельские публикации в сетевых СМИ – их можно считать надежным индикатором маркетинговой активности компаний.

Если говорить о российском рынке, первое, что бросается в глаза – молчание лидеров рынка западного. Конечно, международных новостей о Red Hat и Novell на ресурсах немало, но в России (если судить по публикациям) активность этих компаний близка к нулю.

Удивительный факт, но в апреле в сетевых СМИ «отметились» только компании «Mandriva.Ru» и «Альт Линукс». Первая представила публике корпоративный Mandriva Enterprise Server 5.1 с технической поддержкой на русском языке. Продукт полностью соответствует спецификации LSB 4.0 (Linux Standard Base) и отличается пятилетним сроком поддержки. Покупателям Mandriva Enterprise Server 5.1 предлагаются годовая и трехлетняя подписки.

В «Альт Линуксе» решили не отставать от коллег и объявили о подготовке к выпуску серверного дистрибутива с длительным сроком поддержки (также 5 лет). Альфа-версия ALT Linux 6.0 Centaurus уже доступна для скачивания, а релиз намечен на осень этого года. О коммерческой поддержке речь пока не идет, но, полагаю, для релиза она будет доступна. Еще «Альт Линукс» и «Softline» объявили о партнерстве: подразделению «Softline Linux Solutions» присвоен эксклюзивный статус «Премиум партнер» компании «Альт Линукс». В рамках этого соглашения SLS будет поставлять заказчикам продукты «Альт Линукс», реализовывать проекты на их основе и предоставлять поддержку по этим решениям.



› **Mandriva.Ru вывела на российский рынок свой новый корпоративный продукт.**

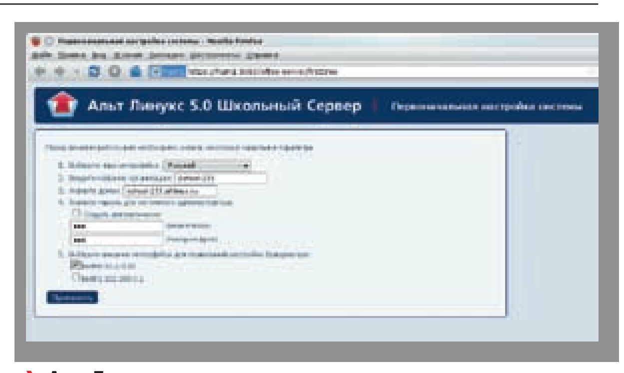
Как видите, новостей по поводу корпоративного сегмента маловато. Видимо, в России даже в этом секторе предпочитают заниматься поддержкой самостоятельно, а высокую активность на рынке проявляют только две компании (периодически возникающие мелкие проекты не в счет).

Linux идет в школу

Удастся ли нам вырастить поколение детей, для которых название операционной системы не будет синонимом слова «компьютер»?

Новостей, касающихся государственных инициатив в области СПО, напротив, предостаточно. СМИ активно обсуждают внедрение Linux в российских школах. На первом месте по частоте упоминаний здесь оказалась компания «Альт Линукс», которая во второй половине марта представила новый комплект школьного ПО. В «Альт Линукс 5.0 Школьный» входят несколько вариантов серверных и клиентских дистрибутивов «Альт Линукс 5.0.1». По утверждению представителей компании, за время модернизации нового комплекта было устранено более ста критических ошибок, добавлено около двух тысяч пакетов, а все программы обновлены до актуальных версий. Главное новшество – увеличение числа программ для образовательного процесса, хотя, на мой вкус, со специализированным ПО дела в школьных дистрибутивах обстоят не очень хорошо. Все диски доступны для бесплатной загрузки; есть и коробочные версии с талоном технической поддержки 20 компьютеров сроком на 1 год. Стоимость такой коробки – 3500 рублей. В «Альт Линукс» всерьез настроены на лидерство в данном сегменте: компания регулярно публикует бесплатные учебные материалы по своим продуктам, проводит тематические семинары и конференции, а число упоминаний «Альт Линукс» в прессе растет в геометрической прогрессии. Компания также стала членом некоммерческого партнерства «Руссофт».

Маркетинговая активность проекта EduMandriva ниже, чем у «Альт Линукс» (судя по публикациям в прессе), но в налаживании связей с вузами проект продвинулся дальше: достаточно посмотреть на перечень центров компетенции. Тактика EduMandriva состоит в работе с ведущими вузами регионов. Очевидно, эти люди разбираются в структуре системы образования: учебные заведения, с которыми они работают – кузницы педагогических кадров для региона, и выбор ПО российскими школами будет во многом зависеть от них. Напоследок скажу пару слов еще об одном дистрибутиве: компания «ЛИНУКС ИНК» представила «НауЛинукс Школа 5.4». Он основан на Scientific Linux 5.4 и полностью совместим с пакетной базой RHEL, CentOS и Scientific Linux.



› **«Альт Линукс Школьный сервер» будет поддерживаться как минимум до 2011 года.**

Видимо, активность российских разработчиков связана с утвержденным президентом Планом реализации Стратегии развития информационного общества в РФ до 2011 года, согласно которому к концу 2010 года 25 % общеобразовательных учреждений каждого региона РФ должны использовать ПСПО не менее чем на 50 % имеющихся персональных компьютеров.

Иллюстрация © www.altlinux.org

Мода на Linux Foundation

Весна-2010 оказалась для организации урожайной: желание вступить в ее ряды выразили сразу несколько компаний.

В апреле к некоммерческому фонду Linux Forundation присоединились сразу три крупных игрока ИТ-рынка. Счет открыла Ricon – производитель принтеров, сканеров, фотокамер и прочего периферийного оборудования. По заявлению руководителя центра разработки компании, вступление в Linux Foundation демонстрирует лояльность Ricon к платформе GNU/Linux. В составе организации она примет активное участие в развитии проекта OpenPrinting.org.

Практически одновременно с Ricon о вступлении в LinuxFoundation объявила Parallels – компания, специализирующаяся на разработке программного обеспечения для виртуализации и автоматизации. В списке ее проектов – коммерческая система виртуализации уровня ОС Virtuozzo и ее свободная редакция – OpenVZ. Продукты Parallels активно используются сервис-провайдерами, хостинговыми компаниями и корпоративными дата-центрами по всему миру. Основателем и руководителем фирмы является Сергей Белоусов. Штаб-квартира Parallels находится в Швейцарии, но главный центр разработки расположен в Москве. Кроме того, практически все высшее руководство Parallels – граждане России. Аманда Макферсон [Amanda McPherson], вице-президент по маркетингу и программам разработки Linux Foundation, по этому поводу заявила: «Parallels является важным дополнением к сложившейся группе участников Linux Foundation. Данная компания действительно приумножает преимущества Linux в виртуализированных средах и понимает цену совместной работы с сообществом». Подобный шаг одного из лидеров в области технологий виртуализации предсказуем – еще до присоединения к организации Parallels входила в десятку самых активных участников разработки ядра GNU/Linux. Сам Сергей Белоусов, правда, «отметился» нелестными эпитетами в адрес свободного ПО, давая интервью немецкому журналу «t3n», хотя позже назвал это попыткой «шутливо указать на существующую тенденцию общего восприятия Open Source и сыграть в адвоката дьявола по отношению к Open Source и сообществу».



› Сергей Белоусов может быть иногда неосторожен в высказываниях, но его компания делает для Linux многое.

Затем ряды Linux Foundation пополнили еще два новых участника: компания Meyer, известная своими акустическими системами для автомобилей, и крупная южнокорейская корпорация LG Electronics. Meyer использует Linux в продуктах линеек D-Mitri™ (цифровое аудио) и Galileo® (система управления акустикой). Основной причиной вступления корпорации LG в организацию стали планы по использованию Linux в качестве одной из основных платформ для производимых ею устройств.

> «Компания Parallels — важное дополнение к Linux Foundation.»

Что было Конференция Softline Linux Solutions. 30 марта, Москва

На конференции, осветившей вопросы использования Linux и свободного ПО в построении ИТ-инфраструктуры предприятия, выступили руководители Softline Linux Solutions и ведущих ИТ-компаний: Novell, Sun, Scalix, VDEL, Cyan-Networks.

Доклады руководителя Softline Linux Solutions (SLS) Николая Здобнова и технического руководителя SLS Тараса Абламского были посвящены комплексным услугам специализированного подразделения компании Softline по внедрению и поддержке программных комплексов на основе Linux и СПО. Большой интерес у аудитории вызвало выступление вице-президента компании Xandros Паскаля Лаурии [Pascal Lauria], который специально прилетел из Германии для участия в конференции. В своей презентации г-н Лаурия открыл возможности нового для российского рынка решения Scalix для организации корпоративной почты. С марта 2010 года продукты Scalix поставляются на территории России и СНГ эксклюзивно компанией Softline.

Представитель компании Sun Виталий Титов, руководитель практики программных продуктов в России и СНГ, представил перспективы развития продуктов Sun в свете слияния Sun с Oracle, которое было анонсировано в начале 2010 года.



› Публике рассказали о перспективах продуктов Sun в свете слияния с Oracle.

В конце мероприятия участники получили сувениры от компании Softline. Один из докладчиков и спонсоров конференции – Денис Морозов, продакт-менеджер компании Cyan-Networks – вручил двум участникам, победителям конкурса анкет, лицензии на годовое использование продукта Cyan Secure Web (для компаний с количеством пользователей 50 и 100 человек, соответственно), предназначенного для обеспечения безопасности ИТ-инфраструктур.

Tech Days 2010

Команда *Linux Format* побывала на главном европейском мероприятии, посвященном Solaris и Java.

8 апреля седьмой павильон «Ленэкспо», что на Васильевском острове в городе Санкт-Петербурге, вновь открыл свои двери гостям Tech Days – ежегодной конференции для разработчиков. Она стала пятой в череде Tech Days, проходивших в России, четвертой – в городе на Неве, и единственной Tech Days в Европе в 2010 году.

В прошлый раз расхожим вопросом, задаваемым едва ли не всем докладчикам, был «Скажите, правда ли, что IBM намерена купить Sun Microsystems?» На что следовал неизменный ответ: «Спросите любого из присутствующих здесь сотрудников компании, и вы услышите: “Sun не комментирует слухи”». За прошедшие 12 месяцев эти слухи обрели форму, слияние с Oracle было одобрено «в верхах» – что же значат все эти изменения для конференции, столь любимой многими (по разным причинам)?

Первое, что бросается в глаза еще до посещения выставочной зоны – это баннер: во-первых, он красно-белый (вопреки ожидаемому сине-зеленому); во-вторых, название мероприятия стало на три буквы короче. Место, где можно смотреть живую видеотрансляцию пленарных докладов, удобно развалившись в кресле-мешке, теперь носит название «Гостиная Oracle». Любители вести досужие споры вокруг того, будет ли ребрендинг, могут извлечь из этого очередной аргумент для своих дебатов; мы же двинемся дальше вглубь демо-зоны и посмотрим, оказала ли новая форма влияние и на содержание.

В этом году программа двухдневной конференции состояла из более чем 40 докладов, традиционно разделенных на три секции: Enterprise Computing, Client Technologies и Solaris. Нововведением стала так называемая стендовая секция, на деле представлявшая из себя четвертый малый зал, в котором проходили выступления на различные специализированные темы.

День первый

Главным разочарованием Tech Days 2010 стало отсутствие изобретателя Java Джеймса Гослинга [James Gosling], чей пленарный доклад должен был открывать конференцию. Организаторов тут винить не в чем: незадолго до начала мероприятия Гослинг ушел из компании (см. ниже), а замена ключевого докладчика – дело непростое и небыстрое. Зато другая «приглашенная звезда» – директор Oracle по продуктам Solaris Дэн Робертс [Dan Roberts] приехал и сделал доклад с многообещающим названием «OpenSolaris: взгляд в будущее». Воодушевленные недавним заявлением в блоге Робертса (гласящим, что Oracle будет и дальше развивать OpenSolaris как открытый проект), но несколько озадаченные отсутствием релиза 2010.03 слушатели заняли в зале секции №3 все места, включая и стоячие. Ничего кардинально нового о будущем OpenSolaris, впрочем, в докладе не прозвучало: г-н Робертс еще раз пробежался по тезисам, которые изложил в своем »



› Какой же Tech Days без традиционной Java Party, скромно названной в программе фуршетом?

Фото: Татьяна Цыганова

Гослинг: «Примите мои извинения»

Джеймс Гослинг покинул компанию Oracle второго апреля. Вот что он пишет на эту тему в своем блоге: «Да, слухи верны: я ушел из Oracle неделю назад (2 апреля). Приношу свои извинения всем в Санкт-Петербурге, кто пришел на TechDays в четверг, ожидая меня услышать. Очень неприятно, что меня там не было. Почему я ушел, ответить трудно: почти все, что можно сказать, принесет больше вреда, чем пользы. Самое, конечно, тяжёлое – это больше не быть со всеми теми замечательными людьми, с которыми я имел честь работать все эти годы. Я не знаю, что буду делать дальше: наверное, сделаю небольшой перерыв перед тем, как искать работу.».

Причины этого увольнения пока не очевидны. Хотя слияние компаний проходит по мягкому сценарию, в Sun идет достаточно серьезная реформа, и спикера, готового комментировать это событие, сейчас найти сложно.

› Живую трансляцию выступлений можно было с комфортом смотреть из Гостиной Oracle.

блоге некоторое время назад, после чего перешел к ключевым возможностям операционной системы, знакомым, пожалуй, каждому, кто имел с ней дело.

Если вы присутствовали на конференции и после первой пары-тройки выступлений у вас сложилось впечатление, что Tech Days 2010 стали менее «Tech», чем в прошлом году, то вы, в общем, не одиноки – открывавшие мероприятие доклады были явно рассчитаны на широкую аудиторию. Нельзя сказать, что это плохо (в чем-то даже наоборот), и тем, кто ожидал большего (включая, разумеется, нас) следовало лишь опуститься с небес на землю – в смысле, перейти со второго этажа на первый, в зал стендовых докладов. Здесь звучали короткие сообщения практически на любой вкус: от сетевого стека *VirtualBox* до Java в спорте.

Для технаря

Желанными гостями на Tech Days 2010 стали специалисты компании JetBrains: хотя ее офис и Центр высоких технологий Sun в Санкт-Петербурге находятся менее чем в 10 километрах друг от друга, разработчики *IntelliJ IDEA* (и не только) участвовали в работе конференции впервые – по крайней мере, на нашей памяти. В двух докладах на стендовой сессии и одном пленарном было рассказано о жизни компании после публикации исходных текстов ее ключевого продукта по лицензии Apache 2.0 (менеджеры других фирм-разработчиков ПО! Если вы читаете это, имейте в виду: продажи не только не падают, но даже растут) и представлена система мета-программирования *MPS* (www.jetbrains.com/mps), которая также является свободным ПО. После ее анонса в прошлом году некоторые интернет-ресурсы решили, что в JetBrains создают монстра, способного программировать за программиста, но на самом деле речь идет лишь о средстве для создания предметно-специфичных языков (Domain Specific Languages, DSL) и языковых расширений – довольно удобном, если, конечно, вас не пугает Java.

С большим интересом мы послушали и другого спонсора Tech Days – компанию Intel. Если вы создаете приложения на C++ и озабочены вопросами производительности (а кто ими не озабочен?), советуем вам обратить внимание на *Threading Building Blocks* (www.threadingbuildingblocks.org) – тем более, что это свободное ПО (GPLv2), а также прочесть обзор на стр. 15.

Конечно, Tech Days не были бы Tech Days без традиционных конкурсов и мастер-классов. На последних нам поприсутствовать, увы, не удалось: участие в них было возможным лишь по предварительной записи; но, скромно постояв в уголке, мы выяснили, что посетители конференции в этом году могли получить навыки практической работы с ZFS, технологиями JavaFX (ключевой темы Sun Tech Days 2009) и *MySQL* и освоить средства обеспечения безопасности Solaris и *DTrace*.



› Изюминкой конференции этого года стала стендовая секция.

Наши впечатления

Каждая конференция Tech Days, как правило, открывается пленарным заседанием, на котором формулируются основные темы этого года. На сей раз их список был несколько размытым – из явных «фаворитов» можно было отметить, пожалуй, только *JDK 7* (Java-программисты, возрадуйтесь: в **switch()** скоро можно будет использовать строки!). И даже беглого сравнения программ текущего и прошлого годов становится достаточно, чтобы понять: фокус сместился. На Tech Days 2010 было несколько меньше Solaris (фактически, большая часть докладов в одноименной секции была или обзорной, как выступление Робертса, или касалась смежных тем: компиляторов *Solaris Studio*, *DTrace* и т. д.); почти не было *Netbeans* (если не считать стенда в демо-зоне) и совсем не было *OpenOffice.org*; зато было довольно много *VirtualBox*. Что это может означать? Мы не можем сказать наверняка: видимо, наиболее заметные и важные изменения за минувший год произошли именно в этих технологиях.

В чем на Tech Days 2010 решительно не было недостатка, так это в посетителях. На момент написания репортажа статистика еще только подсчитывалась, и мы пока не можем назвать вам точные числа, но у нас создалось устойчивое впечатление, что эта конференция стала самой посещаемой среди всех Tech Days последних нескольких лет. С одной стороны, это не может не радовать – хотя в демо-зоне, построенной по «лекалам» прошлого года, было тесновато (особенно в обеденный перерыв). Имейте это в виду, если надумаете посетить мероприятие в будущем году.

Впрочем, доклады Tech Days 2010 можно было смотреть и с комфортом, не выходя из собственной комнаты: их прямая трансляция велась в Интернете. К настоящему моменту на сайте конференции (sun.ru/techdays) уже опубликованы презентации в формате PDF, так что, если какая-то из озвученных тем вас особенно заинтересовала, вы можете загрузить их и во всем разобраться. Ну, а мы, пожалуй, завершим на этом наш краткий обзор. Два дня Tech Days 2010 были насыщенными во всех смыслах, и мы весьма благодарны тем, кто нашел время подойти к стенду *Linux Format*, чтобы поделиться своими впечатлениями или обменяться с нами контактами. До новых встреч – пока на страницах журнала! LXF

LINUX FORMAT

Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов



АЛЕКСЕЙ ФЕДОРЧУК
Его слабости – mass storage, разметка диска и файловые системы.

Незнаменитый офис

Когда речь заходит об открытых и свободных офисных пакетах, вспоминают, как правило, *OpenOffice.org*, реже – вечный долгострой проекта KDE, *KOffice*. И мало кто упомянет в этой связи компоненты *GNOME Office* – текстовый процессор *Abiword* и табличный редактор *Gnumeric*. Что, в общем-то, резонно – обе эти программы вполне самостоятельны и их интеграция достаточно искусственна. Однако это не умаляет их достоинств, каковыми считаются лёгкость, быстродействие, простота интерфейса. Но при этом забывают о функциональности – а ведь каждое из данных приложений обладает своими уникальными особенностями.

Для *Abiword* это средства коллективной работы. Во-первых, он поддерживает мультиверсионные документы – в том числе и те, что были сделаны таковыми в *Microsoft Word*. Во-вторых и главных, *Abiword* располагает инструментами удалённого редактирования: по собственному протоколу AbiCollab.net, прямому подключению TCP и, наконец, по протоколу XMPP – то есть через самый обычный Jabber-клиент.

Ну, а «фирменные фичи» *Gnumeric* – это изобилие статистических и инженерных функций (более ста из которых уникальны) и широчайшие возможности для построения технических диаграмм и графиков.

Иными словами, считать *Abiword* и *Gnimeric* жалким подобием левой руки *OOo* несправедливо. Просто это действительно программы не для офисного клерка, а для инженера, научного работника… в общем, технического специалиста.

alv@posix.ru

Сегодня мы рассматриваем:

KDE 4.4 10
Самый стабильный релиз из вышедших по сей день. Исправлены тысячи ошибок, *Plasma* стала другой, и рабочий стол выглядит по-новому. Удивительно, чего можно достичь за шесть месяцев.

OpenOffice.org 3.2 11
Едва ли не впервые за всю свою историю, самый популярный свободный офисный пакет стал загружаться заметно быстрее. Что это: влияние Oracle или просто удачное совпадение?

PogoPlug 12
Нам нравятся маленькие коробочки, внутри которых оказывается спрятан компьютер под управлением Linux – даже если они розовые. Ждем не дождемся еще чего-нибудь в этом роде.

Denix 2.9.0 13
Создается впечатление, что свой клон Ubuntu за последнее время не выпустил только ленивый. Вот и модный Linux-автор Денис Колисниченко решил попробовать свои силы на новом поприще.

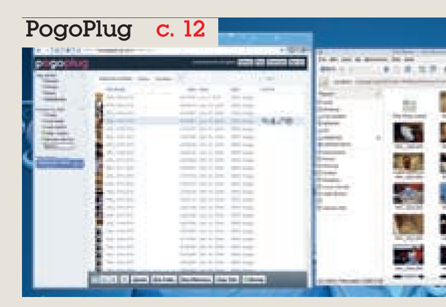
VueScan 14
Взгляните на сканер, который пылится без дела в углу вашей комнаты. Когда-то он не работал, но с этой маленькой да удаленькой программой все может измениться самым коренным образом.

Книги 15
Разрабатываете ли вы web-приложения или пишете код на C/C++, в книжном обозрении этого месяца найдется кое-что для вас: познакомьтесь с *Grails* или оптимизируйте код под многоядерные процессоры.

KDE 4.4 с. 10

› Очередной крупный релиз KDE Software Compilation завоюет себе немало новых сторонников.

PogoPlug с. 12



› Обменивайтесь данными через Интернет благодаря маленькой невзрачной коробочке.

Наш вердикт: Пояснение



Все попавшие в обзор продукты оцениваются по одиннадцатибалльной шкале (0 – низшая оценка, 10 – высшая). Как правило, учитываются функциональность, производительность, простота использования и цена, а для бесплатных программ – еще и документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.

Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но если разработчики рекомендуют *Autopackage*, мы следуем этому совету.

LINUX FORMAT Вердикт

Google Earth

Разработчики: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Оправданность цены	9/10

» Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряющая практичная программа.

Рейтинг 9/10

KDE 4.4



Новый KDE 4.4 – пригоден ли он, наконец, к массовому использованию? **Нейл Ботвик** покинул командную строку и проверил.

Вкратце

» Комплексная среда рабочего стола с набором прикладных программ. См. также Gnome.

В феврале увидела свет версия 4.4 этого неисчерпаемого KDE; пока мы пишем эти строки, его включают в «весенние» дистрибутивы. В LXF124 мы устроили тотальную проверку KDE 4.3 – продолжаем держать вас в курсе событий.

По данным системы контроля ошибок KDE, было исправлено 7293 дефектов и реализовано 1433 запроса на функции. Немало, но для KDE 4.3 эти цифры были куда более внушительными.

Заметнее всего изменение внешности – появились группируемые («вкладочные») окна. Эту функцию необходимо активировать в разделе Appearance [Внешний вид] настроек системы. После этого можно щелкнуть по заголовку окна и выбрать группу, в которую его следует переместить, либо нажать среднюю кнопку мыши и перетащить заголовок окна на заголовок другого – получится два окна в одном. Хотя окна и называются «вкладочными», вкладок как таковых здесь нет: просто в верхней строке появляется два заголовка вместо одного. Экономится экранное пространство, что очень полезно для нетбуков и других устройств с малыми экранами. К сожалению, пока эта функция реализована только в теме Oxygen.

> «Заметнее всего изменение внешности – вкладочные окна.»

Владельцам нетбуков стоит попробовать новый интерфейс *Plasma* – запуск приложений с помощью значков, с панелью для избранных программ, где каждое окно открывается на весь экран. Это один из лучших интерфейсов для нетбуков.



› Новый интерфейс Plasma для нетбуков вместил массу материала в маленький экран, а вкладочные окна удобно организуют рабочее пространство.

Семантический рабочий стол, включивший поиск не только по названиям и датам, но и по содержимому файлов, признан созревшим и интегрирован с *Dolphin*. Теперь по умолчанию в верхней части окна *Dolphin* помещен текстовый поиск по содержимому файлов, занимающий (благодаря использованию базы данных *Nepomuk*) всего несколько секунд. Немного улучшена работа со съемными носителями: всплывающее меню виджета действует быстрее.

Большинство приложений – просто улучшения версий из KDE 4.3, но есть и новинки: например, клиент для ведения блогов *Bloglio*. Функции социального рабочего стола, впервые представленные в KDE 4.3, усовершенствованы. Теперь этот виджет называется *Community*, и сообщения можно отправлять непосредственно из него. Новый виджет *Social News* показывает «живые» обновления.

Для разработчика

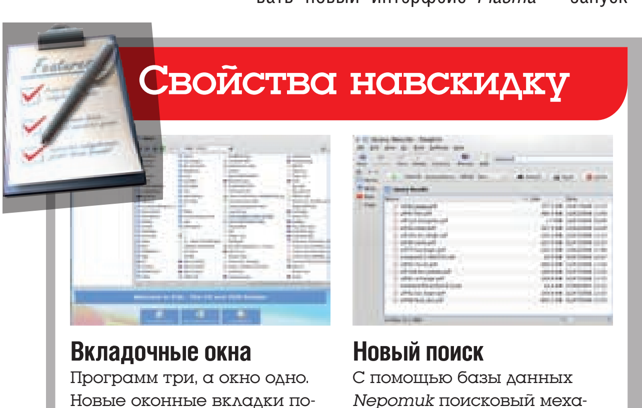
Есть и незаметные на первый взгляд новшества, рассчитанные на программистов. Это переход на *Qt 4.6*, новая среда безопасной аутентификации *KAuth*, централизованное управление цифровой личностью для социального рабочего стола, функции коллективной работы и улучшенная поддержка стандартов.

KDE 4.3 страдал нестабильностью. В итоге *X* «падал» не очень часто, но непредсказуемо, что раздражало особо. KDE 4.4 мы пользовались с момента его появления в бета-версии, и за все это время не было ни одного отказа. Единственная странная неполадка замечена в диалоге *KRunner*. Это окно, появляющееся при нажатии Alt+F2, внезапно «замерзало» на несколько секунд. Все остальное работало, но диалог отказывался принимать команды.

Некоторые компоненты KDE требуют изрядной доли ресурсов, но общая производительность рабочего стола вполне удовлетворительна.

Это крупный шаг вперед. Новые функции не революционны, но повышение стабильности и удобства налицо. Теперь мы можем провозгласить: KDE наконец-то готов для всеобщего применения. LXF

Свойства навскидку



Вкладочные окна
Программ три, а окно одно. Новые оконные вкладки помогают упорядочить работу и сэкономить экранное пространство.

Новый поиск
С помощью базы данных *Nepomuk* поисковый механизм *Dolphin* может искать по содержимому файлов.

LINUX FORMAT Вердикт

KDE Software Compilation 4.4

Разработчик: KDE
Сайт: www.kde.org
Цена: Бесплатно по лицензии GPL

Функциональность	8/10
Производительность	7/10
Простота использования	7/10
Документация	8/10

» Наконец-то KDE 4.x готов к широкому применению и стабилен; но ему еще есть над чем поработать.



Рейтинг 8/10

OpenOffice.org 3.2

Грэм Моррисон нашел золотое дно в, казалось бы, рутинном релизе флагманского офисного пакета мира свободного ПО.

LINUX FORMAT на DVD

Вкратце

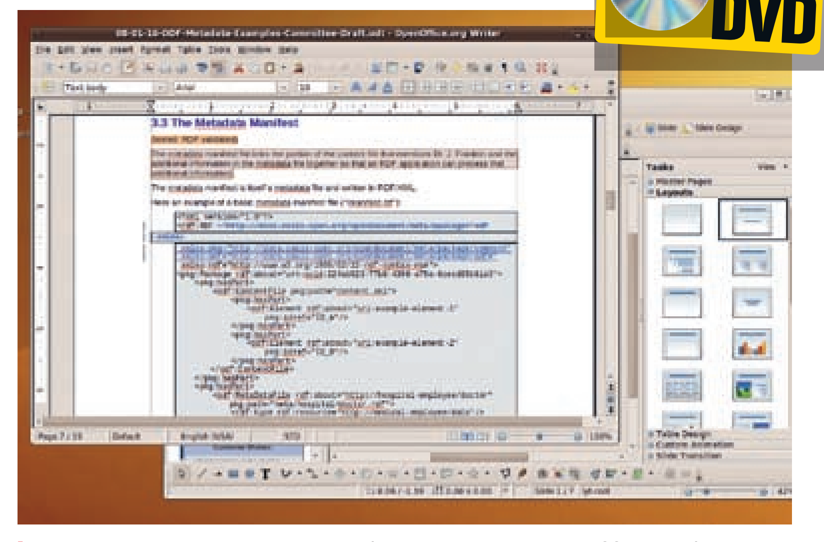
» **Наиболее зрелый офисный пакет, доступный для Linux. См. также:** *KOffice*, *Google Docs* и *AbiWord*.

Гламура в офисных пакетах мало. Они утилитарны, практичны и прозаичны; они напоминают о реальной работе, рутине и обязанностях. Но затем-то они и нужны, и *OpenOffice.org* останется важной частью экосистемы свободного ПО, даже если прекратит обновления.

OOo 3.2 – шаг в правильном направлении. Во-первых, он серьезно ускорился. *Writer* версии *3.2* запускается аж на 50 % быстрее версии *3.1*. В наших тестах время «холодного» старта (сразу после перезагрузки системы) сократилось с 7 до 3,4 секунды. Одно это уже неплохо; но и интерфейс, похоже, стал отзывчивее. Последние две недели мы пользовались исключительно *Writer* и ощутили уменьшение задержки при наборе: для борзописцев – реальное улучшение.

Еще одна забота новой версии – форматы файлов. *OOo* заявляет об улучшенной совместимости с форматами *Microsoft Office 2007* и новой спецификацией ODF 1.2. Но мы, испытав конверсию на некоторых документах *Microsoft Office 2007*, не заметили реального сдвига: она и в версии *3.1* была вполне достойной. Улучшение стоит в том, что теперь можно открывать файлы, защищенные паролями, а также таблицы с OLD-объектами. Файлы **.docx** по-прежнему нельзя ни экспортировать, ни сохранить (несмотря на глухой намек на это новшество в перечне функций).

> «Writer версии 3.2 в наших тестах запускался аж на 50% быстрее.»



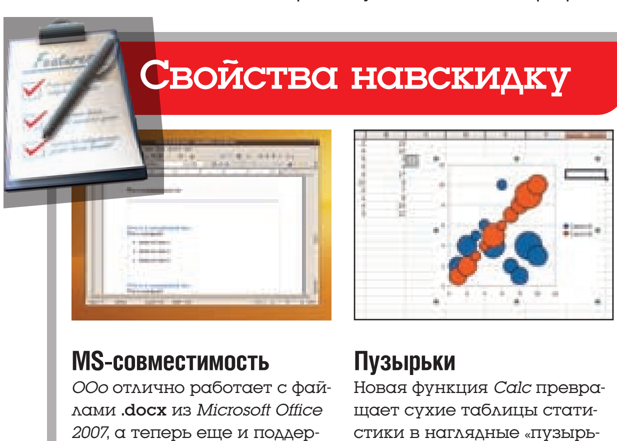
› **Мощное ускорение и новый уклон на работу в корпорациях ставят *OOo* на удобную позицию для обновления интерфейса в русле грядущего ренессанса.**

С файлами Open Document Format *OOo* работает гораздо лучше, а вопрос о совместимости **.docx** может смягчиться, когда (и если) Microsoft наконец интегрирует поддержку в следующую версию своего офисного пакета. Значительно улучшена совместимость со спецификацией ODF 1.2, включая поддержку RDF-метаданных. Это разумное решение направлено на работу с крупными объемами документации.

Курс на корпорации

RDF – сокращение от Resource Description Framework (среда для описания ресурсов). Этот метод обмена данными регулярно применяется в Интернете. Оснащение *OOo* таким средством позволит добавлять к документам метаданные – например, ярлыки и категории – и увеличит совместимость с другими решениями, понимающими RDF. Хороший пример – истории болезней пациентов в клинике. Здесь RDF можно использовать для добавления данных медосмотров, записанных в формате XML. Понять данную концепцию довольно сложно; но такие функции способны привлечь *OOo* к использованию в государственных учреждениях и на предприятиях – а это дело хорошее.

Завершить рассказ о нововведениях просто, поскольку рассказывать особо не о чем. Пожалуй, тезаурус в *Writer* стал «сметливее»: теперь, предлагая слова для ввода, он учитывает словоформы и морфологию – но мы не заметили сколько-нибудь заметной разницы при работе с британским словарем. *Calc* научился генерировать Bubble Charts [Пузырьковые диаграммы], где трем значениям ставятся в соответствие X, Y и область в «пузырьке». Редактор SQL в *Base* обзавелся поиском и заменой. Появилась поддержка шрифтов OpenType на основе PostScript. Вот и все. Да, добавки незначительные, и в общем обновление скромное; но мы в восторге от того, что программисты наконец-то озаботились быстродействием. LXF



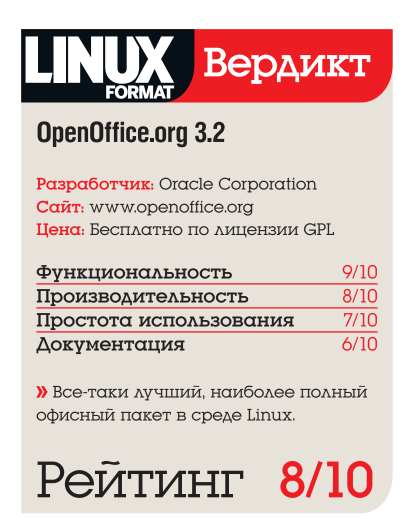
Свойства навскидку

MS-совместимость

OOo отлично работает с файлами **.docx** из *Microsoft Office 2007*, а теперь еще и поддерживает пароли.

Пузырьки

Новая функция *Calc* превращает сухие таблицы статистики в наглядные «пузырьковые» диаграммы.



LINUX FORMAT Вердикт

OpenOffice.org 3.2

Разработчик: Oracle Corporation
Сайт: www.openoffice.org
Цена: Бесплатно по лицензии GPL

Функциональность	9/10
Производительность	8/10
Простота использования	7/10
Документация	6/10

» Все-таки лучший, наиболее полный офисный пакет в среде Linux.

Рейтинг 8/10

PogoPlug

Розовый, красивый, втыкается в маршрутизатор. И за пару шагов создает вам персональное «вычислительное облако». Пробовал **Энди Ченнел**.

Вкратце

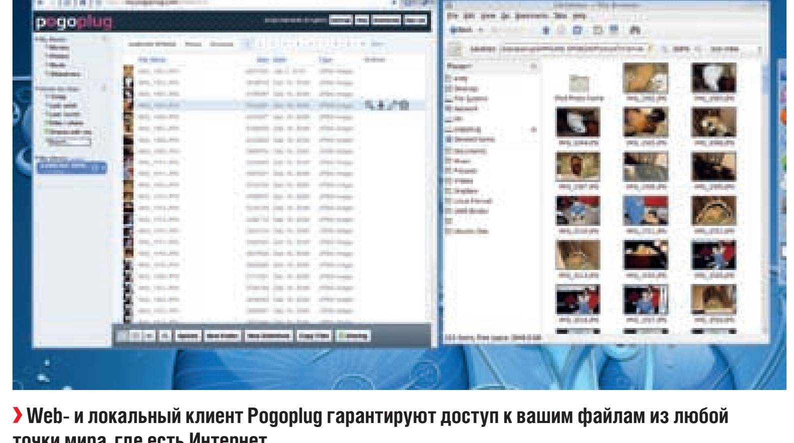
» Сетевой файловый сервер. См. также: *Tonido*, *Sockso* **и** *Opera 10*.

Вычислительное облако – это вещь. Фирмы миллиардами тратятся на разработку сервисов, чтобы уговорить вас доверить свои данные их серверам. А если удобств хочется, но засылать ценные сведения за чей-то брандмауэр боязно? Существуют решения типа *Tonido*, *Sockso* и системы *Unity* от *Opera*, но они работают только в том случае, если ваш ПК постоянно включен.

Pogoplug – это аппаратный облачный сервер со встроенным дистрибутивом Linux. К нему можно подсоединить до четырех USB-накопителей и обращаться к ним через сайт Pogoplug или специальный настольный клиент для Windows, Linux, OS X, iPhone или Android. По сути, сервер монтирует удаленные накопители к вашему рабочему столу. А значит, вы получите доступ к своим файлам везде, где есть Интернет.

Размер устройства – примерно со среднюю книгу в твердом переплете, а настройка до смешного проста: включаете питание, подключаете сервер к маршрутизатору, дожидаетесь зеленого огонька на передней панели и подсоединяете накопители. После визита на сайт Pogoplug для регистрации устройства и установки пароля ваши диски становятся доступными через Интернет, даже за брандмауэром.

> «Вы получите доступ к своим файлам везде, где есть Интернет.»



› Web- и локальный клиент Pogoplug гарантируют доступ к вашим файлам из любой точки мира, где есть Интернет.

Программа-минимум – удаленный доступ к файлам; а еще можно синхронизировать видео, фото и музыку, сохраненные с одной из машин (с установленным настольным клиентом) на Pogoplug при помощи программы *Active Copy*. Эта функция напоминает *Dropbox*, зеркалирующий файлы на нескольких компьютерах. Превосходное решение для резервного копирования и способ организовать удаленную медиа-коллекцию: вы можете размножить содержимое своего диска, затем деактивировать *Active Copy*, и обращаться к файлам через Pogoplug. Поскольку файлы теперь хранятся в сети, свою машину можно от них разгрузить.

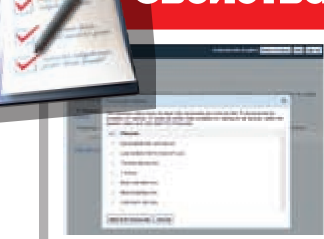
Удаленный доступ к файлам организован здорово, но остается проблема скорости: широкополосные подключения доступны пока не везде.

Быстрый доступ

Устройство авторизуется через сайт Pogoplug, после чего вы подключаетесь непосредственно к серверу, то есть дома вы работаете со скоростью локальной сети – можно сказать, Pogoplug представляет собой очень недорогую NAS-систему.

Решая проблему безопасности данных, Pogoplug создает другую: приходится рассчитывать на то, что маленькая и амбициозная фирма сможет проводить аутентификацию и в долгосрочной перспективе. Вернее, это могло бы быть проблемой для проприетарного ПО. Встроенный дистрибутив Linux позволяет настроить подключение в обход шлюза с помощью одного из пакетов, взятых на www.plugapps.com. Краткий обзор сайта может подсказать вам и другие способы использования устройства. Можно даже разместить пакет на одном из USB-накопителей, чтобы не затрагивать встроенное ПО.

Нам понравились простота устройства, развитые файлообменные возможности, малое энергопотребление и нетребовательность к ресурсам. Конечно, все эти функции можно осуществлять программно, но со значительными издержками. Открытая архитектура придает устройству дополнительную привлекательность, особенно для любителей «аппаратного» хакерства. LXF



Свойства навскидку

Доступ отовсюду

Web-интерфейс позволяет выгружать, загружать и управлять файлами на накопителе с любого сетевого компьютера.

Трансляция видео

Pogoplug транскодирует первые 10 секунд всех видеофайлов, имеющихся на ваших накопителях – но можно обработать их и целиком, для удаленного доступа с iPod.

LINUX FORMAT Вердикт

Pogoplug

Разработчик: CloudEngines
Сайт: www.pogoplug.com
Цена: €99

Функциональность	8/10
Производительность	7/10
Простота использования	9/10
Документация	7/10

» Простой и безопасный метод доступа к информации с различных удаленных устройств.



Рейтинг 8/10

Denix 2.9.0

Грозят ли самому плодовитому русскоязычному Linux-автору лавры Патрика Фолькердинга? Разбиралась **Дина Сенина**.

Вкратце

» Универсальный свободный дистрибутив, совместимый с Debian и основанный на Ubuntu. См. также: Runtu или сам Ubuntu.

Денис Колисниченко, скорее всего, известен нашему читателю как автор многочисленных самоучителей по Linux. Но у нас есть основания полагать, что вскоре он станет популярен еще и как разработчик собственного дистрибутива. Основанный на Ubuntu, Denix Linux – сравнительно компактная (ISO-образ редакции Full занимает всего 1 ГБ) система, появившаяся в виде миниатюрного LiveCD 1 июля 2009 года. Сейчас Denix достиг версии 2.9 (как видите, схема нумерации отличается от родительской). По словам автора, он лишь слегка усовершенствовал «африканца». Но, на наш взгляд, такое преобразование способно облегчить жизнь даже начинающего линуксоида. Denix включает наиболее известные программы с широкой функциональностью.

Работа установщика интуитивно понятна. Все специфические для Denix элементы устанавливаются позже, специальным сценарием. Одновременно удаляются ненужные компоненты. Работа над созданием собственных репозиториев ведется, но покамест Denix ориентируется на официальную пакетную базу Ubuntu.

В основе Denix – два принципа: гибкость (комфортная работа и на сильных, и на слабых машинах) и «одна задача – одно решение». Снабженная эргономичной и приятной графической средой *GNOME*, система содержит несвободные драйверы для видеокарт ATI и NVIDIA, а для ноутбуков и нетбуков предлагается утилита изменения яркости экрана (в редакции Lite).



› Приложения Denix весьма функциональны, а рабочая среда элегантна и эргономична.

Бочка меда

Помимо стандартной связки офисных и интернет-приложений (*Firefox, Thunderbird* и иже с ними), Denix предлагает утилиты для русскоязычного пользователя. Сюда, например, входит *gXneur* – аналог популярного в мире Windows приложения *Punto Switcher*, переключающего при наборе «неправильную» раскладку клавиатуры. С архивами управляется оболочка *PeaZip* – она не только понимает все распространенные форматы, но и умеет разрезать и склеивать несжатые файлы (конечно, это умеют и консольные команды **split** и **cat** – см. LXF130; но не забудем о дружелюбии к пользователю). Есть средства для работы с PDF-документами: от стандартного просмотра в *Xpdf* до редактора *PDFedit*, допускающего автоматизацию из скриптов, и взломщика паролей *PDFCrack*.

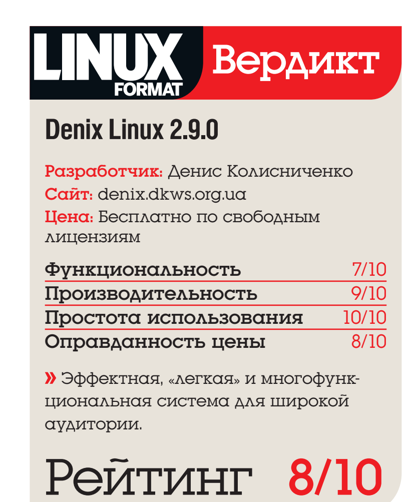
Но Denix отнюдь не чисто конторская система – домашние пользователи также будут приятно удивлены. К их услугам – кодеки и программы для создания и воспроизведения аудио и видео и спецприложение для проигрывания DVD. Denix понимает массу форматов электронных книг, включая DjVu. Модули расширения *Firefox* и свободный проигрыватель *Gnash* обслужат мультимедиа-контент, почерпнутый в Сети, а объемные файлы загрузит менеджер закачек *D4X*. На базе Denix можно даже создать сервер – необходимое ПО приложено.

Ложка дегтя

Увы, никто не совершенен – и дистрибутиву, разрабатываемому одним человеком, простительны некоторые огрехи; однако знать о них все же надо. Например, на этапе установки последний раздел на диске создается только при разметке с конца; учетные записи обычных пользователей можно добавить лишь после установки. Межсетевой экран настраивается в момент инсталляции, но дополнительные средства безопасности все-таки не помешают.

Несмотря на эти проблемы и сравнительно юный возраст, Denix уже получил положительные отзывы и от пользователей настольных систем, и от владельцев ноутбуков и нетбуков. В самом деле, Denix – универсальный дистрибутив; работает «из коробки», содержит упорядоченный и продуманный набор программ, максимально прост в настройке, поддерживает широкий спектр действий с файлами самых разнообразных форматов, закачками и архивами. При необходимости он легко трансформируется в сервер, предусматривающий мониторинг занятости ресурсов системы. Для новичков автор выложил на сайт проекта русскоязычное руководство по установке, «с картинками».

Зачем Денис Колисниченко создал еще один отпрыск Ubuntu? Ну, если звезды зажигают – значит, это кому-то нужно... LXF



VueScan 8.6.10



Ник Вейч глазам своим не верит. Неужели это ПО с ходу распознает любой сканер? Под Linux?

Вкратце

» **Программа для сканирования с поддержкой множества различных устройств. Для работы со сканерами на условиях GPL можно попробовать *XSane* или другое ПО на основе *Sane*.**



Вы можете думать, что к основополагающим правам человека относится право сканировать документы под Linux. Увы, мир против вас. Поскольку перед началом разработки серии 2.6 из ядра были исключены специфические модули, почти каждый экземпляр Linux в наши дни использует для сканирования драйвер *libusb*.

Данное ПО проверяет USB-порт, находит устройство, затем (обычно) перекладывает работу драйвера на плечи микропрограммы сканера соответствующей (если повезет) модели. Несмотря на старания *Sane* (это главный компонент открытого ПО для сканирования), многие сканеры не работают под Linux должным образом – а то и вовсе не работают.

Проприетарное приложение *VueScan* разрабатывается с 1997 года, с простой целью: помочь сканировать все и вся. Интерфейс локализован на многие языки, включая русский, белорусский и украинский.

> «Лицензия Pro дает пользователям бесплатные обновления.»

Программа поставляется в виде подготовленных двоичных архивов, рассчитанных на Ubuntu и Red Hat 9. Возможна установка и на другие дистрибутивы (мы пользовались Fedora), но если у вас возникнут трудности – официально поддерживаются только эти два. Зависимостей у приложения немного, и работать оно должно на любой машине – что относится и к поддерживаемым сканерам.

Для теста мы взяли Epson 3170 Photo. Теоретически он должен работать под Linux, но за несколько месяцев возни с *Sane* из сканера удалось извлечь лишь пару всхлипов. *VueScan* мгновенно распознал устройство, а через пару минут радостно сканировал документы. На сайте программы (http://bit.ly/B14rs) приведен внушительный список поддерживаемого оборудования – но учтите, что устройства, требующие наличия *CyberView X*, в Linux не поддерживаются.

Итак, одна из причин использовать *VueScan* – высокая совместимость с оборудованием. Но это не все. Возможности управления процессом сканирования уму непостижимы. Параметров так много, что по умолчанию большинство из них скрыты – не то растеряешься! Для рядового пользователя оно и лучше: по-быстрому отсканировать пару документов можно и без тонкой настройки.

Уровень профи

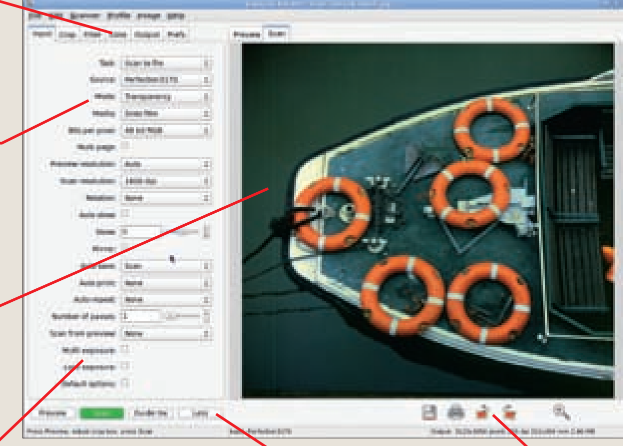
Однако будьте уверены, что для серьезной работы по сканированию найдется все. *VueScan* поддерживает ICE (нечто вроде «прочистки» сканера), если ваш сканер с этим работает, управляет цветом и автоматически генерирует цветовые мишени IT8.

Для сканирования прозрачных материалов есть несколько тонких настроек, но самая интересная из них – фиксация выдержки. На примере маленького участка ракорда программа определяет необходимую выдержку, а затем применяет ее ко всей пленке. Если это цветные негативы, то выполняется корректировка цветов. Оптическое распознавание тоже неплохое.

Но и на солнце бывают пятна. Скажем, последовательное сканирование весьма трудоемко. Оцифровывая отрезки кинопленки, придется повозиться с настройкой смещений и промежутков. Дескрининг выполнять тоже непросто – существуют утилиты (правда, для Mac и Windows), справляющиеся с этой задачей лучше.

В общем, даже стойкое предубеждение против платы за ПО в этом случае может быть поколеблено. Лицензия Pro дает пользователям право пожизненно получать обновления бесплатно. А другое ПО для сканирования вам вряд ли когда-либо понадобится. LXF

Интерфейс VueScan



Вкладки Settings
Управление всеми параметрами, от коррекции цветов до выходного формата, кроется здесь.

Тип оригинала
VueScan работает с носителями различных типов, включая прозрачки, негативы и кассеты.

Панель предпросмотра
По мере сканирования в зоне предпросмотра появляется копия сканируемого изображения.

Фиксация выдержки
Фиксированная выдержка упрощает обработку партии негативов или печатных материалов.

Упрощенный интерфейс
Кому кажется, что параметров перебор, может перейти в автоматизированный интерфейс попроще.

Быстрый доступ
Часто используемые инструменты группируются здесь – для ускорения доступа.



VueScan 8.6.10

Разработчик: Эд Хамрик [Ed Hamrick]
Сайт: www.hamrick.com
Цена: $40 (версия Pro — $80)

Функциональность	9/10
Производительность	9/10
Простота использования	10/10
Оправданность цены	9/10

» В Linux приложению просто нет равных — а может, и на других платформах тоже.


Рейтинг 9/10

Grails. Гибкость Groovy и надежность Java

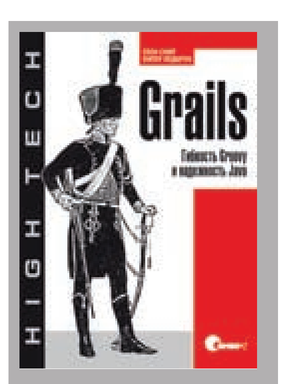
Платформа Java известна не только своей надежностью, но и громоздкостью. Тем не менее, она пригодна и для быстрой разработки web-приложений.

Конечно, с этим утверждением можно поспорить, но мы склонны считать, что платформа *Ruby On Rails* (*RoR*) действительно изменила мир web-разработки. По крайней мере, с ее появлением в головах широкой общественности укоренилась мысль, что не всякий web-сайт необходимо писать с нуля. Нет, разумеется, библиотеки существовали и раньше, но *Rails* подняла эту идею на новый уровень.

Grails является собой попытку скрестить лучшее из *Rails* с Groovy, быстроразвивающимся скриптовым языком для платформы Java. Не являясь точной копией *RoR*, *Grails* тем не менее заслуживает внимания всех, кто хочет создавать web-приложения на Java быстро и без лишнего кода.

Аннотация гласит, что данная книга написана «практиками и для практиков». Ее авторы, Глен Смит и Питер Ледбрук, стояли у истоков *Grails* и создавали с его помощью web-проекты, когда он находился еще в версиях 0.x. Вы можете рассматривать их труд как пошаговое руководство по web-разработке.

Книга состоит из четырех частей. Первая знакомит читателя с платформой *Grails 1.1* на примере простого приложения (в принципе, если вас интересует только обзор технологии, ею можно и ограничиться). Здесь также даются основы языка Groovy (версии 1.6), что делает «Grails ...» единственным русскоязычным изданием о нем, по крайней мере, до тех пор, пока не выйдет перевод «Groovy in Action». Последняя часть охватывает продвинутые вопросы, вроде внутреннего устройства *Grails* и технологий *Spring* и *Hibernate*, лежащих в его основе. Все, что находится между ними, являет собой практический пример реализации современного web-приложения: от определения модели данных до интерфейса Web 2.0 и тестирования. Жаль, что интернационализация при этом упоминается лишь походя.



› «Гибкость Groovy и надежность Java» — не вполне верный перевод для «... in Action», но заявленную тему книга раскрывает.

Весьма бодрый темп изложения, избранный авторами, делает книгу не слишком подходящей для новичка – но ветераны web-разработки благодаря ей и расширят кругозор, и получат в свое распоряжение практически-ориентированное руководство по *Grails*, причем весьма полезное. LXF

LINUX FORMAT Вердикт

Grails. Гибкость Groovy и надежность Java

Авторы: Глен Смит, Питер Ледбрук
Издательство: Символ-Плюс
ISBN: 978-5-93286-185-1
Цена: 750 руб.
Объем: 656 стр.

» Неплохой способ разобраться с практическим применением платформы Grails, если вы опытный web-разработчик.

Рейтинг 9/10

Оптимизация ПО. Сборник рецептов

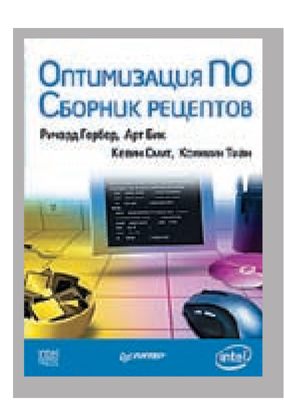
Процессор с четырьмя ядрами бесполезен, если три из них простаивают. Узнайте, как загрузить их работой, но при этом не перенапрячь.

Когда-то для ускорения работы программы было достаточно купить ей компьютер помощнее. К сожалению, эти времена прошли, и сегодня, чтобы заставить ваш код выполняться быстрее, его необходимо оптимизировать. А поскольку большая часть процессоров, установленных в современных компьютерах, многоядерные (а бывают и многопроцессорные компьютеры), все, что касается параллелизма, здесь выходит на первый план. Основная задача этой книги, написанной специалистами компании Intel, в том и состоит, чтобы подсказать разработчику направление движения в сторону оптимального кода.

Книга содержит три части. В первой, озаглавленной «Инструменты и концепции повышения производительности», читатель знакомится с тем, как измерить производительность работы программы и определить в ее коде места, нуждающиеся в оптимизации. Этот разговор завершает краткий обзор архитектуры современных микропроцессоров.

Вторая часть является самой объемной: в ней последовательно раскрываются препятствия на пути производительной работы вашего приложения (алгоритмы, неэффективные с точки зрения параллельного выполнения; неверно предсказанные переходы и попадания мимо кэша процессора; циклы и потенциально медленные операции) и излагаются стандартные способы их обхода, включая OpenMP (но не Threading Building Blocks – их первая версия вышла почти одновременно с книгой), и методики, специфичные для процессоров Intel. В третьей части все изученное сводится в единое целое на практическом примере – видеокодеке H.264.

Книга предполагает читателя с соответствующим уровнем подготовки: как минимум, вы должны свободно читать ассемблерный код. Оригинал издания вышел в 2006 году, поэтому вы не найдете здесь информации о новейших процессорах. Авторы ориентируются на инструменты Intel (в большинстве своем закрытые и платные) и (в основном) на платформу Windows, но изложенные в книге методы являются универсальными и с некоторыми изменениями могут быть перенесены на практически любой инструментарий по вашему выбору. LXF



› Ассемблерный код на обложке намекает, что для оптимизации программ нужно хорошо понимать работу процессора.



Оптимизация ПО. Сборник рецептов

Авторы: Ричард Гербер, Арт Бик, Кевин Смит, Ксинмин Тиан
Издательство: Питер
ISBN: 978-5-388-00131-3
Цена: 352 руб.
Объем: 352 стр.

» Ценный источник информации для разрабатывающих серьезные приложения, но требует и серьезного подхода к чтению.

Рейтинг 8/10

Сравнение



Каждый месяц мы сравниваем тысячи программ – а вы можете отдыхать!

LINUX FORMAT на DVD
- Digikam 1.1.0
- Fotoxx 9.7
- F-Spot 0.6.1.5
- KPhotoAlbum 4.1.1
- Rawstudio 1.2
- RawTherapee 3.0 a1

Фотоменеджеры

Ник Вейч ищет способ разложить снимки по полочкам...



Про наш тест...

Кроме отдельно оговоренных случаев, все программы устанавливались на Fedora 12 с двухъядерным процессором на 3 ГГц и 4 ГБ ОЗУ. Для каждой использовался один и тот же набор файлов – смесь JPEG и RAW-снимков с «зеркалки» Nikon. Остальные файлы импортировались с помощью стандартного кард-ридера.

Мы пробовали провести простейшее редактирование, по возможности преобразовать RAW-файлы и сохранить или выгрузить результаты. Учитывалось также потребление памяти. Там, где было можно, мы пытались скорректировать изображения – например, устранить шум (на снимке Луны с высоким значением ISO) и сделать поправки на линзу. *Bibble 5 Pro* содержит урезанную версию программы *Noise Ninja*, также доступной как полноценный модуль расширения. Полную версию мы не тестировали.

Наш выбор

Fotoxx	с. 17
Picasa	с. 17
Bibble Pro	с. 18
Rawstudio	с. 18
F-Spot	с. 19
Raw Therapee	с. 19
Digikam	с. 20
KPhoto Album	с. 20

С выходом цифровых камер в народ жесткие диски стали заполняться фотками почти с каждого события жизни, скучного или веселого. Счастливые обладатели аппаратов мигом уяснили, что если раньше в поисках нужного снимка достаточно было глянуть в ящик стола или фотоальбом, то теперь на диске они исчисляются тысячами. Так и возникла потребность в программах-менеджерах фотографий.

Цифровая технология привела к тому, что люди многократно снимают одно и то же, успевай щелкать; итог – избыток похожих изображений. Хорошие программы должны помочь отсеять ценные фотографии от тех, чье место в мусорной корзине, отображая качественные миниатюры и отслеживая источники. Также необходимы тэги и метаданные.

> «Хорошие программы должны помочь отсеять ценные фотографии.»

Цифровые камеры класса high-end обычно выдают так называемые RAW-изображения. Это данные высокого качества, взятые прямо с сенсора камеры. Но они приводят к неимоверно большому количеству типов файлов: ведь сам по себе RAW лишь термин, применяемый к бездне форматов – как правило, проприетарных. Так как ресурсы камеры ограничены, то создаваемые ими файлы JPEG можно улучшить только с помощью настольной программы. Хорошее фотоприложение должно понимать данные RAW и преобразовывать их в JPEG. Для этого часто применяется испытанная утилита *dcraw*, и все наши программы декодируют RAW-файлы либо через нее, либо через собственные библиотеки, с переменным успехом.

Хорошие организационные функции подразумевают возможность снабжать фотографии тэгами, извлекать из камеры метаданные и, наконец, выгружать, отображать или печатать избранные снимки. Одни люди просто держат свои фотки на компьютере, а другим тут же хочется поделиться ими с миром. Поддержка фотосайтов является бонусом, даже для профессиональных программ. Большинству же требуется какое-никакое хранение отснятого материала и поиск понадобившегося.

Fotoxx

Эта темная лошадка таит удивительные возможности.

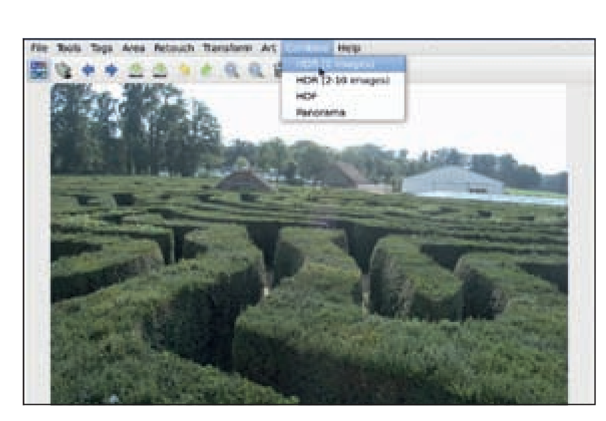
F*otoxx*, хотя и стала знаменитой, попав несколько раз в HotPicks, далека от главного русла программ управления фотографиями с точки зрения функций и, возможно, пользователей. Но ее стоит принять в расчет. Например, она поддерживает RAW-изображения и содержит набор утилит для управления ими. Есть здесь и функции, отсутствующие в ряде других рассмотренных нами программ.

Большая часть этих экзотических функций прячется в меню Combine [Скомбинировать]. Например, режим Panorama [Панорама] комбинирует несколько снимков с минимальным ручным вмешательством. Снимки помещаются один за другим, и результат очень неплох – это не замена программам-сшивателям, делающим полный монтаж, но получается настоящая панорама 360°. То, что *Fotoxx* называет режимом HDF – похожее комбинирование: берется несколько снимков с разными областями в фокусе, и они собираются в непрерывное целое, с бесконечной глубиной резкости.

Есть множество других трюков, типа ручного тонирования и поддержки HDR. Опять-таки, предусмотрена ручная подстройка, и нет проблем со снятием отдельных изображений самостоятельно, вместо поисков камеры, умеющей фотографировать «по три за раз». При поддержке массы RAW-форматов, благодаря библиотеке *UFRaw*, и 16-битной глубине цвета, *Fotoxx* может ковать высококачественные изображения, получив правильный исходный материал.

Базовая поддержка тэгов реализована на основе данных Exif, но она децентрализована. Некоторые могут поспорить, что такой способ организации изображений лучше – *Fotoxx* сохраняет ваши тэги и прочую информацию прямо в Exif. Недостаток данного подхода – при поиске конкретной фотографии требуется просмотреть все изображения, и для начала надо примерно представлять себе, где она находится.

> «Есть множество других трюков, типа поддержки HDR.»



› *Fotoxx* отлично умеет комбинировать несколько изображений.



Fotoxx 9.7

Цена: Бесплатно
Сайт: http://tinyurl.com/fotoxx

» Прекрасна для экспериментов; но, вероятно, не для организации большой коллекции.

Рейтинг 5/10

Picasa

Google пытается обработать изображения по уму.

Если кто не в курсе, *Picasa* – это Google'овский клиент управления фотографиями, прочно увязанный с одноименным фотохостингом. Он взаимодействует с онлайн-службой, однако работает и как автономная программа, даже без учетной записи в Picasa Web Albums.

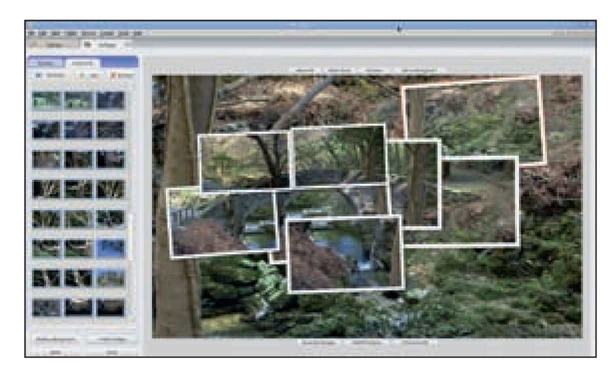
Мы выбрали версию 3.0 Alpha для Linux, что может показаться несправедливым, если учитывать ошибки. Но, как это ни забавно, на большинстве дистрибутивов она работает лучше стабильной версии, 2.7. Самая большая проблема *Picasa* – зависимость от *Wine*. Для Linux это не родное приложение – сборки RPM и Deb более или менее такие же, что и версия для Windows, но идут со встроенной специальной редакцией *Wine*. Краткий обзор форумов скажет вам, что она не всегда ведет себя хорошо.

Если отвлечься от этого, сила *Picasa* – в способности управления и индексирования. При первом запуске программы она разыщет все изображения в вашем домашнем каталоге и разберет их по папкам, упорядоченным по времени. Она быстро импортирует файлы из других источников, и даже снабжена неплохой поддержкой всевозможных сканеров.

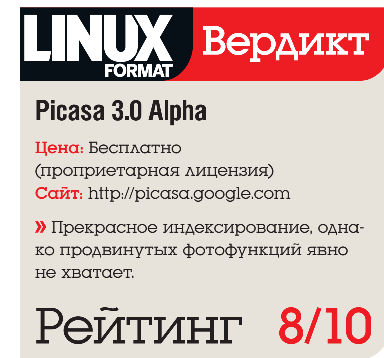
Другая сильная сторона *Picasa* – организация. Она ведет базу данных, однако оставляет ваши изображения на своем месте, предпочитая индексировать их и их тэги. Можно искать по тэгам, и в программе даже есть такая функция, как поиск по цвету. Некоторые опции могут вас озадачить, и Collage [Коллаж], режим грубой сшивки, не для серьезного монтажа, зато быстр и прост в использовании, как и базовые функции редактирования.

Вы можете экспортировать и синхронизировать ваши снимки с онлайновой Picasa, или размещать изображения в блог-сервисе Blogger, однако службами не от Google пользоваться нельзя.

> «В программе даже есть такая функция, как поиск по цвету.»



› *Picasa* хороша как менеджер, а в остальном довольно-таки простецкая.



Picasa 3.0 Alpha

Цена: Бесплатно (проприетарная лицензия)
Сайт: http://picasa.google.com

» Прекрасное индексирование, однако продвинутых фотофункций явно не хватает.

Рейтинг 8/10

Bibble Pro

Функций много, но за денежки отчего не постараться.

Это единственное приложение в нашем Сравнении, которое просит вас растрясти кошелек. Но при виде предлагаемых им функций устоять нелегко. Каталогизация, тэги, цветовые профили, цветокоррекция и коррекция линз... Есть даже база данных линз: ею можно воспользоваться, взяв Exif со своей камеры и выбрав подходящие настройки.

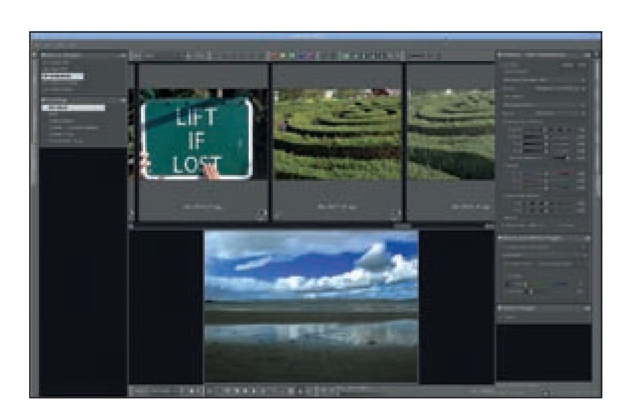
Наличие нескольких баз данных очень удобно, и практически все допускает сохранение в виде профиля, применимого к другим снимкам. Этому способствует каталогизация: она отфильтрует сходные изображения, основываясь на данных Exif.

Впрочем, маловато инструментов для работы с деталями – для легкой ретуши изображений после их экспорта может понадобиться внешний редактор. В оправдание скажем, что если вы готовы потратить некоторое время, в *Bibble* есть искусный режим слоев – применяя инструменты выделения, можно создать настраиваемый корректирующий слой для любого изображения.

Если у программы и есть недостатки, то это мелкий, кособокий шрифт интерфейса, затрудняющий прочтение большинства диалогов и меню. Единственная опция в настройках его только уменьшает!

По цене *Bibble* вполне конкурентоспособен на профессиональном рынке, где, на других платформах, он соревнуется с *Adobe Lightroom* и прочими. В этом плане стоимость не чрезмерна. Он дает гораздо более точные инструменты, чем любое отдельное приложение Linux, и вы получаете коммерческую поддержку.

Возможно, это и слишком для домашнего пользователя, если не замахиваться на такие функции, как устранение шума. Планируется версия Lite, однако данные о ее возможностях и стоимости на момент написания доступны не были.



› Инструменты управления цветом — один из козырей *Bibble*.

> «Дает более точные инструменты, чем любое приложение Linux.»



Bibble 5.0 Pro

Цена: $200
Сайт: http://bibblelabs.com

» Один из наиболее прецизионных инструментов, но за деньги.

Рейтинг 9/10

Rawstudio

Элегантно-серый, но не силен в организации и редактировании.

Rawstudio порожден необходимостью обрабатывать в Linux файлы RAW. Он использует прекрасные библиотеки *dcraw*, поддерживающие уже более 300 камер, и обычно справляется с делом лучше, чем программы, поставляемые изготовителями фотоаппаратов. Он встраивается в рабочий процесс фотографа и нацелен на RAW-изображения (при желании можно загружать также файлы JPEG).

Укажите ему каталог (изображения прямо с камеры не скачиваются, если она не подключена как устройство), и на верхней панели автоматически появятся миниатюры. Щелкните на какой-нибудь, и инструменты на правой панели активируются и помогут в изменении гистограмм цветов, контроле экспозиции и других обычных действиях, потребных для обработки RAW-файлов. Нажатие цифровой клавиши 1, 2 или 3 сгруппирует изображения или расположит их по приоритетам, облегчив обработку и выбор победителя сессии. К одному из трех профилей установок можно применять индивидуальные изменения, а результаты экспортируются и по одному, и пакетом. Это лучшее свойство *Rawstudio*, в особенности для обработки множества изображений.

С точки зрения редактирования, здесь нет ничего, кроме обрезки и распрямления. То же касается функций управления, хотя тэги для рейтинга присутствуют. Правда, *Rawstudio* можно применить как часть рабочего процесса, а потом загрузить сгенерированные изображения для правки в другую программу, например, *GIMP*. *Rawstudio* аккуратно и эффективно генерирует JPEG-файлы и даже обеспечивает их пакетную обработку, но не является для ваших фотопотребностей решением на все случаи жизни.



› Пакетная обработка RAW-изображений — сильная сторона программы.

> «Результаты экспортируются и по одному, и пакетом.»



Rawstudio 1.2

Цена: Бесплатно
Сайт: http://rawstudio.org

» Хорош для пакетной обработки, но для редактирования понадобится что-нибудь другое.

Рейтинг 5/10

F-Spot

Живой пример поколения Mono не лишен прекрасных свойств.

Эта программа явилась ниоткуда, и заслуженно стала приложением по умолчанию для управления снимками в Gnome. Ее подход – обеспечить полное решение для контроля за фотографиями, от скачивания изображения с цифровой камеры до отправки его по электронной почте, печати или выгрузки на один из полдюжины поддерживаемых *F-Spot* онлайн-сервисов (например, Flickr, Facebook и Picasa).

Тэги можно проставлять на момент импорта, а изображения копировать в централизованное хранилище, где они тасуются по папкам в зависимости от даты. По всему каталогу можно перемещаться с помощью верхней временной шкалы (напоминающий *iPhoto*), однако есть и поиск по тэгам. *F-Spot* содержит массу приятных функций, типа поиска дублирующихся изображений, а также хранит историю редактирования, кадрирования и ретуширования файлов.

Проблема в том, что инструменты коррекции не идеальны. Хотя работа по импорту RAW-изображений выполняется хорошо, здесь нет тонкой подстройки кривых, как в *Bibble* или *Rawstudio*. Адекватная регулировка для нормальных изображений имеется, но инструментов редактирования маловато, так что ретушь придется выполнять на стороне. Правда, *F-Spot* запустит для вас приложение и по вашей просьбе сделает резервную копию оригинала.

Иерархические тэги хороши и просты в управлении. *F-Spot* вполне прилично их экспортирует, и вы можете иметь те же тэги, например, во Flickr. Он также генерирует HTML-галерею непосредственно из снимков, для экспорта на вашу личную web-страницу.

Часть полезных функций *F-Spot* реализована через модули расширения, и не все они включены по умолчанию – так что проверьте меню Edit > Manage Extensions [Правка > Управление расширениями].

> «Он хранит историю редактирования и ретуширования файлов.»



› *F-Spot* – прекрасный органайзер, ускоряющий поиск фотографий.



LINUX FORMAT Вердикт

F-Spot 0.6.1.5

Цена: Бесплатно
Сайт: http://fspot.org

» Неплохой универсал, но детальное редактирование придется выполнять на стороне.

Рейтинг 8/10

RawTherapee

Новичку есть чем похвастать.

RawTherapee пока не избалован вниманием пользователей Linux – возможно, потому, что он был выпущен под GPL совсем недавно. Автор решил, что помощь других кодеров пойдет программе на пользу; но нельзя не отметить, что для одного-единственного разработчика результат достигнут поразительно высокий.

RawTherapee попадает в один лагерь с *Rawstudio* и *Bibble*, в том смысле, что это инструмент фотографа с высококачественной «зеркалкой», и если вы не привычны к обработке снимков, обилие органов управления и функций может смутить. Уж очень много этих органов! В этом смысле *RawTherapee* практически не уступает *Bibble Pro*. Может, кое-чего недостает в тонкости и глубине цветовой коррекции или устранении шума, но все функции на месте.

Нас расстроило отсутствие редактора кривой управления контрастностью и яркостью изображения. В ранних версиях программы он имелся, поэтому вероятен его скорый возврат. Как и в других приложениях нашего Сравнения, здесь допускается сохранение различных профилей, которые намного упрощают обработку целых сессий и снимков с определенной камеры.

Как и *Rawstudio*, это программа поддерживает режим пакетной обработки, и задачи преобразования можно запускать в фоновом режиме. На выходе поддерживаются JPEG и 8- или 16-разрядные TIFF'ы и PNG.

Программа пока находится в стадии альфа, и часть функций заморожена – например, просмотр миниатюр; однако, судя по всему, недочеты вскоре будут устранены.

> «Предназначена для пакетной работы с качественными DSLR.»



› Столь тщательный контроль изображения преобразит ваши фотографии.



LINUX FORMAT Вердикт

RawTherapee 3.0 Alpha

Цена: Бесплатно под GPL
Сайт: www.rawtherapee.com

» Потрясающая программа, хотя, пожалуй, сложновата для домашнего пользователя.

Рейтинг 8/10

Digikam

Претендент из KDE выдает впечатляющие изображения.

Как любое приложение в возрасте нескольких лет, *Digikam* пережил и взлеты, и падения; однако последние 12 месяцев – сплошной подъем. Количество добавленных функций просто потрясает. И неудивительно, что перед самым Рождеством вышел знаковый релиз 1.0. Новые возможности включают загрузку 12- и 16-разрядных RAW-изображений и работу в этой разрядности – без потери информации. Это поднимает рейтинг *Digikam* и ставит его в один ряд с *Bibble* и *RawTherapee* по качеству вывода.

Руки приложены не только к технической стороне. Одним из хронических критических замечаний к *Digikam* была сложность его освоения. Отчасти данная критика все еще в силе – он работает не как большинство других фотоприложений, и некоторая крутизна кривой обучения неизбежна. Однако достигнуты успехи в плане дружелюбия к пользователю. Исправлено прежнее расположение снимков «до» и «после» рядышком, и теперь они сменяются по наведению курсора мыши, что позволяет видеть изображение в целом более четко.

Интеграция с *Marble* позволяет выполнить геолокацию, просто перетащив снимок на глобус. Если вы не помните, где сохранили изображение, но знаете, как оно выглядит, всегда можно попробовать нечеткий поиск – начертите на фотографии ваш снимок, и *Digikam* поищет для него совпадения.

Может, это и не самая простая в использовании программа – немало ее лучших функций устанешь искать; всевозможных инструментов настолько много, что все сразу не одолеть. Более новые релизы имеют проблемы со стабильностью, но, к сожалению, причина здесь кроется в библиотеках KDE более низкого уровня. На некоторых дистрибутивах, например, SUSE, работа более надежна.

> «Количество добавленных функций просто потрясающее.»



› **Организационные функции и поддержка 16-разрядности выводят его в победители.**



Digikam 1.1.0

Цена: Бесплатно
Сайт: http://unity-linux.org

» Сочетание функций и великолепное качество результата сулят высокое место.

Рейтинг 9/10

KPhotoAlbum

Организуйте альбомы и экспортируйте их куда угодно.

KPhotoAlbum в общем-то не претендует на большее, чем быть просто программой для просмотра и управления снимками, и играет в паре с *Digikam* в команде KDE. Фактически, вся работа по подготовке фотографий к загрузке проделывается в графических библиотеках KDE: в частности, *KPhotoAlbum* переложил на них поддержку данных Exif.

Один из сразу заметных недочетов *KPhotoAlbum* – он с ходу спрашивает, где вы будете хранить все ваши снимки. Программа индексирует только один каталог, и все изображения обязаны обитать в одном и том же месте. Также отсутствует понятие импорта изображений с камеры или кард-ридера – придется копировать их в свою коллекцию самим.

По части организации, коль скоро изображения добавлены, *KPhotoAlbum* может много чего. У него есть временная шкала, подобно *F-Spot*; он умеет упорядочивать изображения на основе тэгов или свойств. Слайд-шоу также создаются легко и просто.

KPhotoAlbum читает RAW-файлы, благодаря вездесущей утилите *dcraw*, но можно велеть ему не делать этого – есть даже возможность игнорировать RAW-снимки при наличии файла JPEG с тем же названием.

Редактирование ограничивается простыми преобразованиями, но правым щелчком мыши можно отослать снимок другой программе.

Основное достоинство *KPhotoAlbum* – его взаимодействие с Web. В нем, похоже, самый широкий спектр опций экспорта – в основном благодаря поддержке всех основных сайтов фотохостинга, а также Smugmug и Facebook. Это делается через впечатляющий набор модулей расширения, и есть надежда на появление новых.

> «Поддерживает все основные сайты фотохостинга и Facebook.»



› **Благодаря впечатляющей поддержке тэгов и данных Exif это хорошая программа-органайзер.**



KPhotoAlbum 4.1.1

Цена: Бесплатно
Сайт: www.kphotoalbum.org

» Работа с web-ресурсами хороша, но в остальном весьма ограничен.

Рейтинг 4/10

Фотоменеджеры

Вердикт

Digikam 9/10

Во время тестов все программы работали хорошо, даже альфа-версии – то есть они, как минимум, пригодны к использованию. Ключевой вопрос – насколько возможности программ отвечают потребностям среднего фотографа; и кстати, кто этот средний фотограф?

Разные люди проделывают со своими фотками разные действия. Если вы отчаянно рветесь к самым профессиональным функциям и не постоите за ценой, то для вас лучшим выбором станет *Bibble 5 Pro*, по причине полноты его набора возможностей. Он не так сильно дружит с Linux, как другие программы, да и недостаток настроек внешнего вида некоторых раздражает, однако результаты у него отличные.

Значительно дешевле обойдется почти тот же набор функций в *RawTherapee* – у него превосходный инструментарий для высококачественной цифровой фотографии. Однако он скорее тяготеет к профессионалам, а домашние пользователи ощутят недостаток простоты применения и функций для снижения трудозатрат. Для более общего использования, *Picasa*, *F-Spot* и *KPhotoAlbum* имеют свои достоинства, особенно в плане организации. Если вы уже делитесь своими фотографиями онлайн в Picasa, клиент *Picasa* подставит плечо без долгих раздумий. *F-Spot* хорош для организации снимков и поможет в поиске изображений, если вы храните их на своем компьютере, а также справится с выгрузкой и генерацией web-галерей. *KPhotoAlbum* несколько ограничен, так как принуждает вас хранить все фотографии в одном месте, зато у него широкий спектр опций выгрузки в Сеть.

Возможно, причиной стала новая с иголочки версия, но лавры достались *Digikam*. Он сочетает функции органайзера и редактора и свободно управится и с профессиональными снимками, и с семейными фото. Полезные инструменты редактирования и удобные функции автокоррекции почти оправдывают титул фотоприложения-все-включено. Но это не самая простая программа для понимания и использования, а значит, она далека от идеала, и никого не удивит, если вы предпочтете более дружелюбные варианты, типа *Picasa* или *F-Spot*. LXF

> «Digikam управится и с профи-снимками, и с семейными фото.»



› Организация фотографий в *Digikam* всегда позволит найти самые ценные картинки.

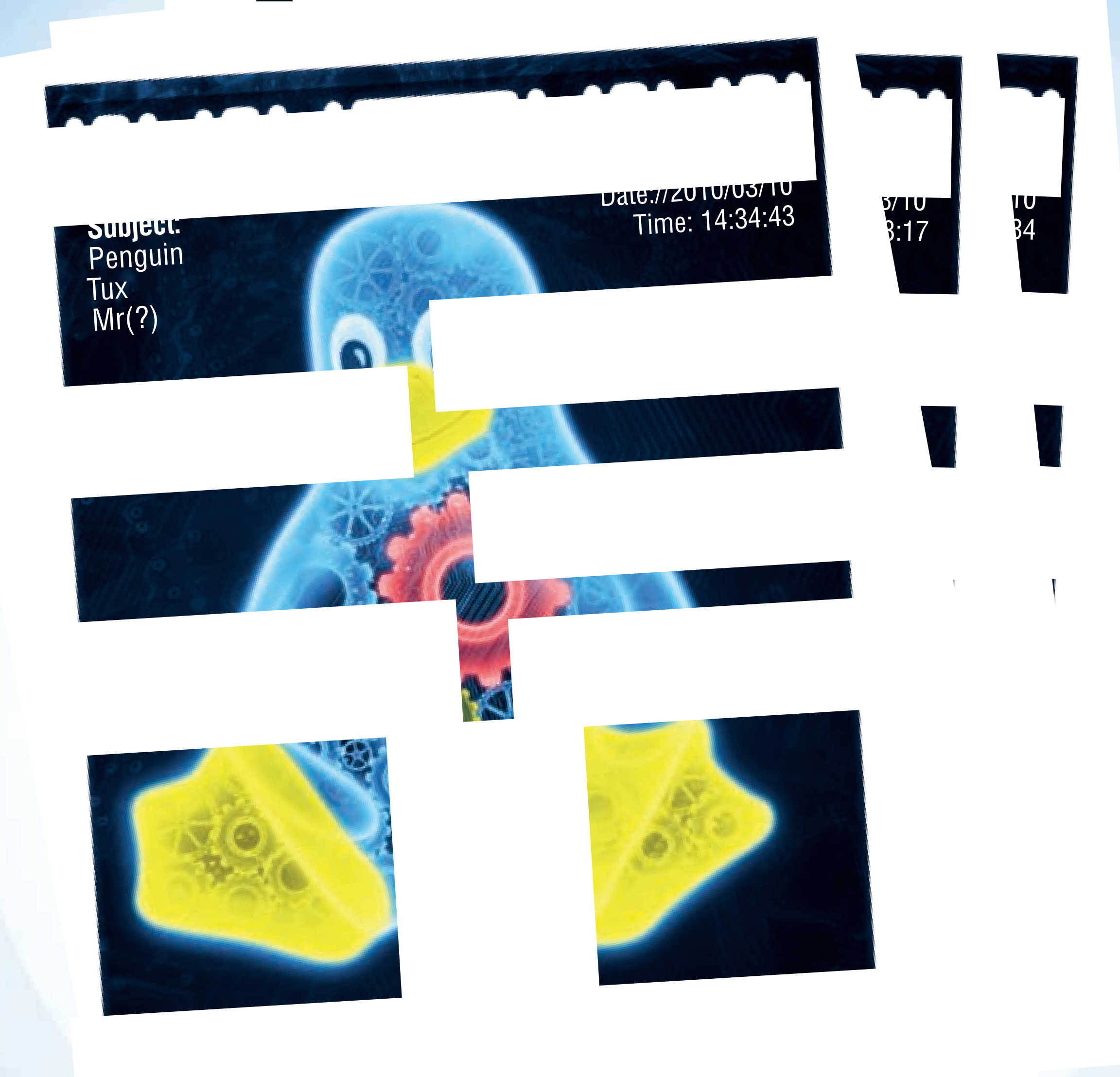
Обратная связь

Подходит ли *Digikam* под ваши нужды? Что для вас главное в программе-менеджере снимков? Мы будем рады услышать ваше мнение – присылайте свои сообщения на letters@linuxformat.ru.

Сравнительные характеристики

Название	Bibble 5 Pro	Digikam 1.1.0	Fotoxx 9.7	F-Spot 0.6.1.5	KPhoto Album 4.1.1	Picasa 3.0 Alpha	Rawstudio 1.2	Raw Therapee 3.0 Alpha
Цена	$200	Бесплатно/GPL	Бесплатно/GPL	Бесплатно/GPL	Бесплатно/GPL	Бесплатно/пропр.	Бесплатно/GPL	Бесплатно/GPL
Память	52 МБ	104 МБ	29 МБ	59 МБ	41 МБ	68 МБ	72 МБ	138 МБ
Инструментарий	Qt	KDE/Qt	GTK	Mono/GTK	KDE/Qt	Wine	GTK	GTK
Поддержка RAW	Встроенная	dcraw	dcraw	dcraw	dcraw	Встроенная	dcraw	dcraw
Цветовые профили	✔	✔	Базовый	Базовый	✘	Базовый	✔	✔
Загрузка на сайты	✘	нет	нет	✔	✔	Picasa/Blogger	✘	✘
Создание HTML-альбомов	✘	нет	✘	нет	✔	✔	✘	✘
Тэги	✔	✔	✔	✔	✔	✔	✘	✔
Рейтинг	✔	✔	✘	✔	✘	✔	✘	✔
Коррекция линз	✔	✔	✘	✘	✘	✘	✔	✔
Цветокоррекция	✔	Базовая	Базовая	Базовая	✘	Базовая	✔	✔
HDR	✘	Базовый	Ручной	✘	✘	✘	✘	✘
Сшивание	✘	✘	Ручное	✘	✘	Базовое	✘	✘

Вскроем Linux



Вы когда-нибудь задумывались о том, что такое DCOP, или где прячутся ваши драйверы? Погрузитесь с **Грэмом Моррисоном** в недра Linux в поисках ответов.

Основная проблема попыток забраться под капот Linux и понять, как он работает, состоит в неосведомленности, с чего начать. Перед вами – огромный пласт сложного ПО, разрабатывавшийся тысячами программистов. Разумно было бы рассмотреть последовательность загрузки и разобраться, что же такое делает *Grub*, а уж потом приниматься за инициализацию RAM-диска и загрузку ядра.

Но недостаток этого подхода очевиден. Достаточно упомянуть о *Grub* слишком рано в какой-нибудь статье – и вы, скорее всего, спугнете большинство читателей. Применение хронологического подхода вызовет ту же проблему и при объяснении принципов работы Linux.

Вместо этого, мы рассмотрим Linux «послойно», продвигаясь «сверху вниз» по технологиям, от рабочего стола до ядра, с позиций «среднестатистического» пользователя. Таким образом мы постепенно перейдем из комфортной зоны рабочего стола вглубь, в «археологию» Linux. Мы обнаружим множество реликтов из давно ушедших времен многопользовательских систем, «тупых» терминалов, удаленных соединений и хакеров былого. Кстати, это одна из причин, почему изучение Linux столь интересно: можно точно увидеть, что произошло, почему и когда. Это позволяет «препарировать» операционную систему так, как было бы невозможно с ее альтернативами. И вы действительно осознаете, почему некоторые вещи «на поверхности» работают так, а не иначе.

Уровень 1 Пространство пользователя

Прежде всего разберемся с базовыми принципами.

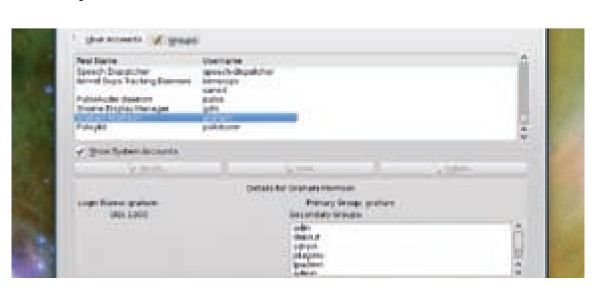
Перед погружением в недра Linux важно понять главную идею. Это – концепция, которая связывает пользовательское пространство [userspace], привилегии [privileges] и группы [groups] и управляет работой и поведением всей Linux-системы в целом, а также взаимодействием с ней пользователей. Концепция эта предполагает, что обычный пользователь не может внести в систему серьезных изменений, не доказав, что он обладает правами администратора, которые это позволяют. Потому-то вам и предлагают ввести пароль, когда вы пытаетесь установить новые пакеты или открываете панели управления конфигурацией вашего дистрибутива, и потому-то обычный пользователь не может видеть содержимого каталога **/root** или модифицировать некие файлы.

В любом дистрибутиве для доступа к настройкам в масштабах всей системы потребуется либо команда **sudo**, либо учетная запись с правами администратора. **Sudo** обычно работает в пределах одного сеанса или команды, и применяется в качестве ситуативного [ad-hoc] решения для повседневного использования, аналогично подходу других ОС, например, Windows 7 и Mac OS X. Правда, войдя в полноценную учетную запись администратора, легко нечаянно там засидеться и забыться так, что допустить роковую ошибку. Но цель у обоих методов одна – безопасность.

Система пользователей, групп и назначения им полномочий организована в Linux для максимальной безопасности. Основная идея – дать пользователям полную свободу действий над собственными файлами, но не допускать их к общесистемным файлам и настройкам, или уж потребовать для этого ввод пароля. Если вы – единственный пользователь вашей системы, такой подход может показаться избыточным. Однако, как мы увидим далее, эта концепция – дань временам, когда в операционной системе имелось множество пользователей и один-два администратора.

Linux – вариация на тему Unix, который многие десятилетия являлся одной из самых распространенных многопользовательских ОС в мире. А значит, многопользовательских функций в Linux избежать трудно; но это и одна из причин его популярности – ведь многопользовательские ОС просто обязаны быть защищены, и Linux унаследовал многие преимущества этих ранних решений.

Например, в Linux учетная запись пользователя автономна. Все ваши персональные файлы хранятся в вашем домашнем каталоге; то же справедливо и для остальных пользователей системы. Имена пользователей, имеющих право доступа к данной системе, можно найти, просмотрев содержимое папки **/home** из менеджера файлов. В зависимости от прав доступа, иногда даже можно заглянуть в домашние папки других пользователей. Но только сам владелец файлов определяет, кому дается к ним доступ, а кто его лишен.



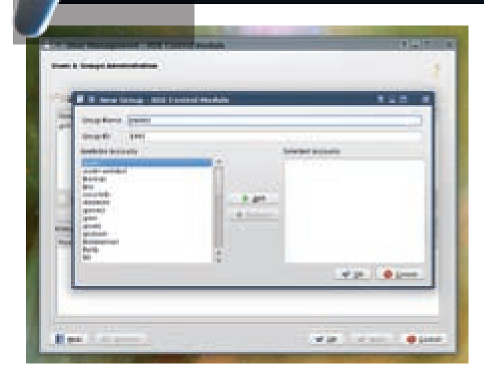
› **Применяя группы, можно активировать или блокировать те или иные сервисы для отдельных пользователей.**

Полномочия

В файловой системе Linux у каждого файла и папки есть девять атрибутов, используемых для задания прав доступа к нему. Эти атрибуты указывают, может ли пользователь, группа или кто угодно читать файл, писать в него или запускать его на выполнение. Пусть вы хотите поделиться своими фото с прочими пользователями системы. Тогда создайте группу 'photos', добавьте в нее пользователей, по вашему мнению, достойных лицезрения, и выдайте этой группе права на доступ к папке – больше никто ваших снимков не увидит. Такая задача решаема в любом современном файловом менеджере: обычно достаточно будет выбрать папку и отредактировать ее свойства [Properties].

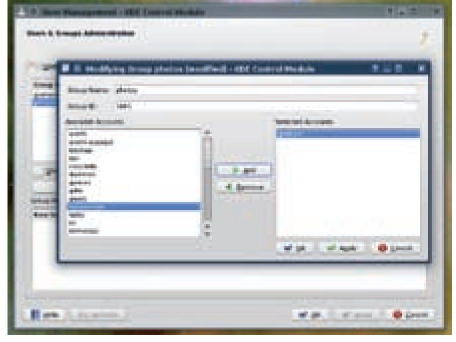
Именно так рабочий стол хранит информацию о настройке ваших приложений, инструментов и утилит. Обычно внутри домашней папки создаются скрытые папки (их имена начинаются с точки); туда помещаются текстовые файлы, где ваш рабочий стол и приложения хранят свои настройки. Другие пользователи их не видят. Тем, среди прочего, и хороша идея переноса всей вашей домашней папки на новый дистрибутив – ведь тогда вы сможете сохранить свои настройки даже при полной замене ОС. »

Шаг за шагом: Присоединимся к группе



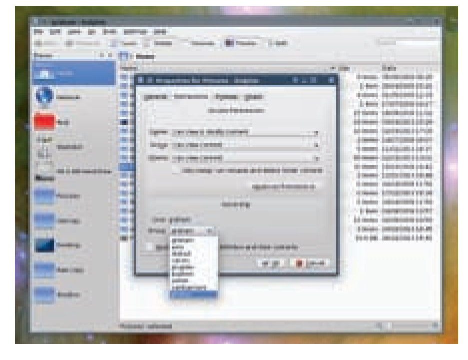
1 Создадим группы

От имени администратора, откройте утилиту User Manager, переключитесь на страницу Groups и добавьте новую группу.



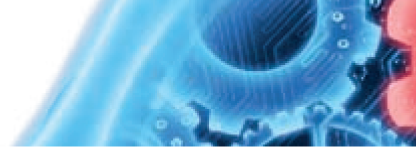
2 Добавим пользователей

Добавьте в созданную группу желаемых пользователей. Возможно, вам придется выйти из системы, а затем снова войти.



3 Отредактируем файл

Теперь задайте через свой файловый менеджер свойства папки, которую вы хотите предоставить новой группе.



Уровень 2 Рабочие столы

Если Linux глубок, как Байкал, то рабочий стол – его сверкающая гладь.

> **Что такое... ACPI**
>
> Это часть вашей системы, которая управляет электропитанием, обычно через системную BIOS. Если не получается нормально выключить компьютер, перевести его в спящий режим или перезагрузиться – ищите проблему именно здесь.

Тем, кто попал в Linux не из серверного зала, а из мира Windows или Mac OS X, попытка объяснять, что такое рабочий стол, покажется диковатой. Все равно что объяснять «чайнику», что Microsoft Windows – операционная система; а онто думал, что это компьютер!

На самом деле рабочий стол представляет собой особый тип приложения, служащего посредником между конечным пользователем и другими программами, которые он может запускать на своем компьютере. Это очень важно, потому что рабочий стол всегда должен «знать», что и где происходит. Только тогда станут возможны такие интеллектуальные вещи, как виртуальные рабочие столы, минимизация окон приложений или разделение их по видам деятельности («занятиям»). Рабочий стол помогает осуществлять эти задачи двумя способами.

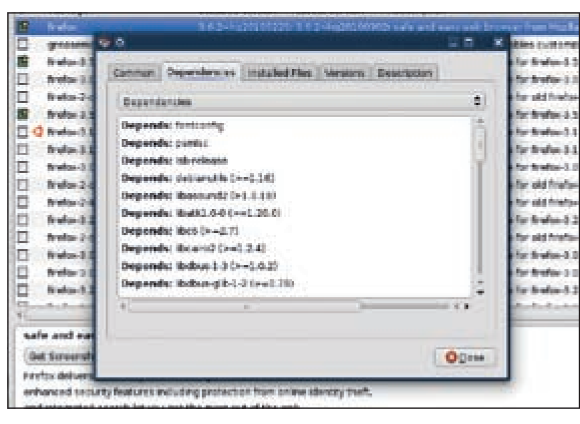
Первый из них – так называемый программный интерфейс приложения (Application Programming Interface, API). Когда программист разрабатывает приложение, используя API рабочего стола, он получает все преимущества, предлагаемые рабочим столом: например, проверку орфографии или создание списка контактов для передачи другому приложению, основанному на том же API. Когда множество приложений использует один и тот же API, создается намного более однородная и непротиворечивая среда, чего мы и ожидаем от рабочих столов вроде Gnome и KDE. Скажем, *K3b* (http://www.k3b.org/) так хорошо работает с вашими музыкальными файлами, поскольку использует тот же API KDE, что и ваш музыкальный проигрыватель; то же относится и к приложениям Gnome.



Инструментарии

Приложения, предназначенные для конкретной среды рабочего стола, не обязаны использовать исключительно какой-то единственный API. Различных API, пожалуй, даже больше, чем дистрибутивов Linux, и они умеют делать все – от сложных математических выкладок до общения с аппаратурой. Здесь-то и звучат термины наподобие *Clutter* или *Cairo*: это дополнительные инструментарии, позволяющие программистам разрабатывать более унифицированные приложения. Например, при создании красивых интерфейсов пользователя с аппаратным ускорением графики для устройств с низким энергопотреблением в Ubuntu Netbook Remix и Moblin используется *Clutter*. Именно он обеспечивает прокрутку верхней панели в Moblin и эффекты «проявления» стартового меню в UNR. *Cairo* облегчает программистам создание векторных графических изображений, и по умолчанию используется как движок визуализации в *GTK*, базовом инструментарии рабочего стола Gnome, для большинства его значков. Векторная графика не привязывает изображение к конкретному разрешению экрана и допускает бесконечное масштабирование, благодаря чему идеальна для изображений, предназначенных для широкого диапазона устройств.





› **Самодостаточные приложения редки — большинство заимствует функции у других программ, так называемых «зависимостей».**

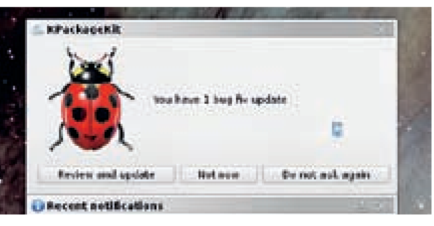
Межпроцессная коммуникация

Второй способ, которым нам содействует рабочий стол – межпроцессная коммуникация. Из названия этой технологии следует, что она помогает процессам «общаться» друг с другом; в среде рабочего стола это обычно взаимодействие между приложениями. Благодаря ему рабочий стол представляет собой единое целое. Например, ваш программный медиа-проигрыватель может узнать о подключении MP3-плейера; или, скажем, ваше ПО для работы с беспроводной сетью пользуется системой уведом-

Обновления

Все основные дистрибутивы имеют свои каналы обновлений и инструменты оповещения о необходимости их установки: благодаря им загрузка и развертывание обновления нередко выполняется одним-двумя щелчками мыши. В системе обновлений Linux замечательно то, что обновление одной-единственной библиотеки исправит проблему со всеми приложениями, которые ее используют. Тем самым управление обновлениями существенно упрощается.

Исключением из этого правила является ситуация, когда вы берете неофициальные пакеты. Ваш дистрибутив может должным образом поддерживать только пакеты, протестированные и выпущенные его кураторами [maintainers]. Часто это оплачиваемые специалисты, и задача проверки, подходит ли тот или иной пакет для включения в официальный дистрибутив, воспринимается разработчиками систем вроде Fedora очень серьезно. Если вы используете свой Linux-компьютер для критически важных задач, необходимо ограничиться только официально поддерживаемыми пакетами. Менеджер пакетов вашего дистрибутива установит их по умолчанию, и вы можете быть уверены, что при наличии проблем они будут исправлены. Именно это происходит, когда менеджер обновлений выскакивает с извещением о том, что один или несколько пакетов следует обновить. Обязательно дайте ему выполнить эту операцию.



› **Лучший метод поддержания безопасности и стабильности системы — регулярные обновления.**

Если вы – обычный пользователь настольного ПК, то, скорее всего, устанавливаете программы из самых разных источников. Например, коммерческие игры нередко поставляются как статически скомпонованные исполняемые файлы. Однако наиболее частый способ установки сторонних пакетов – добавление неподдерживаемого репозитория программ в состав системы управления пакетами вашего дистрибутива. Обычно это репозитории, содержащие новые приложения и их зависимости, но они редко предлагают такой же уровень сопровождения и стабильности, как официальные. Впрочем, в защиту их поставщиков следует сказать, что проблемы чаще всего имеют место и на других платформах, и репозитории типа PPA дают отличный метод протестировать новое ПО – по крайней мере, если вы способны справиться с отрицательными последствиями.

лений об обнаружении новой сети. В общем, межпроцессная коммуникация – причина того, что приложения *GTK* лучше работают в среде Gnome, а приложения KDE лучше работают в KDE. Но главное – то, что оба рабочих стола используют для межпроцессной коммуникации один и тот же совместимый метод: систему под названием D-BUS.

Так почему же тогда Gnome и KDE так непохожи? Отчасти потому, что используют разные оконные менеджеры. Идея оконного менеджера берет начало еще с тех лет, когда Unix-подобные системы только зародились из «первичного бульона» командных строк и начали отображать сеансы терминала в окнах. Можно было перетаскивать эти окна курсором по заштрихованному фону, открывать другие терминальные сеансы и так же манипулировать ими с помощью штуки, известной как *TWM* (Tom's Window Manager). Умел он немногое, но все же освобождал пользователей от текстовых страниц. Окна можно было свободно перемещать по экрану, менять их размеры, распахивать на весь экран и размещать друг поверх друга.

То же делают и современные оконные менеджеры из Gnome и KDE. Оконный менеджер KDE, называемый *KWin*, дополнил возможности *TWM* рядом продвинутых функций: например, встраиванием любого окна в рамку, снабженную вкладками, привязкой приложений к заданной области экрана и еще более специфической привязкой конкретных приложений к собственным виртуальным рабочим столам. Кроме того, *KWin* создает массу сложных эффектов: дрожание окон, отбрасывание теней и отражений и т.д. – идея, впервые воплощенная в *Compiz*, еще одном оконном менеджере, который добавил не функции, а красоты динамического интерфейса в ранее статический мир оконных менеджеров. *Compiz* до сих пор является заменой по умолчанию для оконного менеджера Gnome (*Metacity*), и вы получите его на машине Gnome, активировав расширенные эффекты на панели Visual Effects. Вы увидите, что он плавно замещает стандартные процедуры отрисовки изображений аппаратно-ускоренными.

> «Другие ОС связывают приложения с ресурсами статически.»

Зависимости

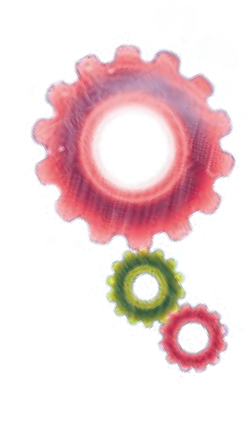
Одно из серьезнейших затруднений для людей, перешедших на Linux – то, что нельзя просто загрузить исполняемый файл из Интернета, чтобы он сразу заработал. Например, при выходе новой версии *Firefox* не получится просто скачать файл с www.mozilla.org, сохранить его на рабочем столе, дважды щелкнуть мышью и установить новую версию. Некоторые дистрибутивы близки к этому идеалу, но проблема остается. Она зависит от конкретного дистрибутива, и мы сейчас не ближе к решению, чем были 10 лет тому назад. Проблема коренится в зависимостях и в разнице способов, которыми дистрибутивы пытаются ее преодолеть.



› Moblin и UNR применяют *Clutter* для аппаратного ускорения графики в устройствах с низким энергопотреблением, типа нетбуков.

Зависимость [dependency] – просто пакет, необходимый приложению для корректной работы. Обычно это API, примененные разработчиками при создании приложения, и они необходимы, потому что приложение использует часть реализуемых ими функций. Когда они упакованы таким образом, их называют библиотеками; приложение заимствует из библиотеки пару-тройку компонентов и добавляет их в состав своей функциональности. Скажем, *Clutter* является зависимостью и для Moblin, и для UNR, и его надо установить, чтобы оба этих рабочих стола могли работать. И хотя, например, *Firefox* выглядит самодостаточным приложением, на самом деле он имеет внушительный список зависимостей, в число которых входят *Cairo*, набор шрифтов TrueType и даже аудиодвижок.

Решая эту проблему, другие ОС связывают приложения с требуемыми ресурсами статически, то есть упаковывают в один файл все, что может потребоваться программе. Все зависимости скрыты в файле **setup.msi** (Windows) или DMG-файле (Mac OS X), дающем приложению или утилите все, что им надо для работы, безо всяких дополнений. Основной недостаток этого подхода в том, что в итоге в вашей системе появляется несколько версий одной и той же библиотеки. Во-первых, это отнимает дисковое пространство, а во-вторых, при обнаружении бреши в системе безопасности придется обновлять все приложения, а не одну библиотеку. »



Шаг за шагом: Межпроцессная коммуникация в деле



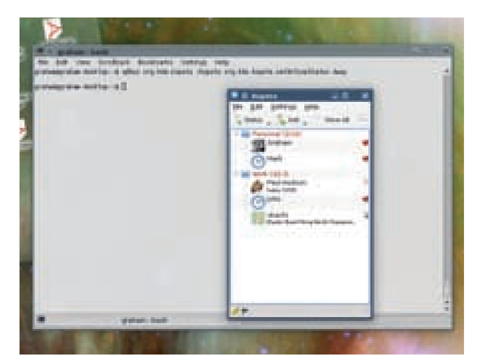
1 Исследуем Kopete

Если вы запускаете *Kopete* в KDE, введите команду **qdbus org.kde.kopete**, чтобы просмотреть все данные, которые каждый из процессов может найти о *Kopete*.



2 Покопаемся в функциях

Введите ту же самую команду, за ней пробел, а далее – любую из строк вывода, полученных на предыдущем шаге, чтобы спуститься на уровень глубже.



3 Сделаем нечто полезное

Попробуйте ввод следующей строки: **qdbus org.kde.kopete /Kopete org.kde.Kopete.setOnlineStatus Away**. D-BUS изменит ваш статус на «Отсутствует».



Уровень 3 Под поверхностью

Покинем безопасную графическую среду, чтобы понять, как работает Linux.

> **Что такое... Inotify**
>
> *Inotify* работает на уровне ядра и уведомляет остальные компоненты системы, когда происходит нечто важное. В основном *Inotify* используется настольными поисковиками, чтобы отслеживать изменение, добавление или удаление файлов.

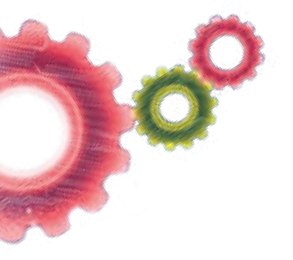
X – не самое умное название для системы, отвечающей за отрисовку окон на вашем экране и управление клавиатурой и мышью, но мы к нему привыкли. Как и серия имен языков программирования: B, C, C++ и C#, оконная система *X* получила такое имя потому, что ее предшественница называлась *W* (в чем все же имелась логика). Буквально с рождения Linux, оконная система *X* стала одним из важнейших компонентов операционной системы. Ее часто критикуют за сложность и размеры, но на свете очень мало программ, которые продержались бы почти 20 лет, особенно с учетом того, как сильно изменились за это время графика и GUI.

Самую большую путаницу вызывает не странное название *X*-сервера, а термины «клиент» и «сервер». Это взаимоотношение напоминает о временах, когда Linux еще не существовало, а оконная система *X* разрабатывалась для дешевых, тупых дисплеев и клавиатур, подключавшихся к мощным машинам с Unix. Машина выполняла всю тяжелую работу по вычислению содержимого окна и созданию GUI. Дисплею оставалось отобразить эти данные и обеспечить взаимодействие. Для гарантий, что средства коммуникаций не окажутся привязанными к единственному поставщику, был разработан открытый протокол для обмена данными между различными устройствами, в результате чего и появилась система *X*.



Перепутаны клиент и сервер

Здравому смыслу в данном случае противоречит то, что сервером является как раз терминал, устройство с экраном и клавиатурой, а клиентом – большая машина, где вся работа возлагается на CPU. В обычной среде клиент–сервер дела обстоят с точностью «до наоборот» – там сервером считается более мощная машина. В системе *X* все перевернуто с ног на голову потому, что именно терминал доставляет ресурсы пользователю, а приложения потребляют их как клиенты.



> «X разрабатывалась для дешевых дисплеев при машинах с Unix.»

› Исходный клиент *XTerm* — по-прежнему резервный терминал для многих дистрибутивов, включая Ubuntu.

Теперь, когда и клиент, и сервер работают на одном компьютере, это больше не проблема. В наши дни настройка производится почти полностью автоматически, но вы все же можете эксплуатировать клиент-серверную архитектуру системы *X*. Например, благодаря такой структуре *X* можно иметь несколько графических сеансов на одной машине. И по той же причине ОС Linux хорошо показала себя в работе с удаленными рабочими столами.

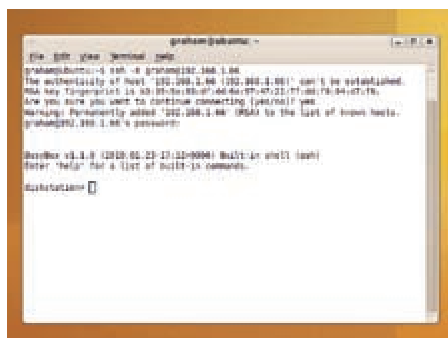
Служба, управляющая аутентификацией при вашем входе в систему, называется PAM (Pluggable Authentication Modules, Подключаемые модули аутентификации). Как намекает ее название, PAM реализует множество типов схем безопасности за счет применения разных модулей. Аутентификация в этом смысле предоставляет способ защиты ваших регистрационных данных, гарантирующий их сравнение с информацией, хранящейся в файлах настройки, так, чтобы при этом процессе данные не подсмотрели и не скопировали. Если модуль PAM не может завершить процесс аутентификации, значит, реквизитам доверять нельзя. В большинстве ди-

Шаг за шагом: Туннелирование X через SSH



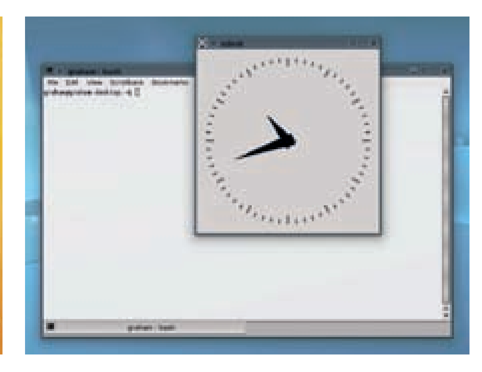
1 Изменим конфигурацию

Убедитесь в том, что в файлах настройки SSH на обоих ваших компьютерах активизированы **AllowX11Forwarding** и **ForwardX11**.



2 Установим соединение

Чтобы установить соединение с удаленным компьютером, введите команду **ssh -X имяпользователя@ip-адрес**. В идеале, на этом компьютере не должно быть запущено сеансов *X*.



3 Запустим графический сеанс

Для запуска графического сеанса через сеанс SSH, введите команду **startx --:1** или попробуйте команду **xclock &**, если такой сеанс уже имеется.

Виртуальные файловые системы

Файловая система Linux – штука необычная. Она может представлять собой смесь локальных и удаленных файлов, работающих процессов и аппаратных устройств, и способна ошеломить новичка. Например, там нет папки 'Program Files', а все ваши личные файлы хранятся в соответствующем каталоге в **/home**. Приложения и библиотеки размещаются в разных директориях, обычно в дереве **/usr** или в **/lib**, но даже эти стандарты могут меняться в зависимости от дистрибутива. Самую серьезную проблему представляют файлы настройки. Они обычно находятся в каталоге **/etc**, но их названия и содержимое зависят от дистрибутива и меняются от версии к версии.

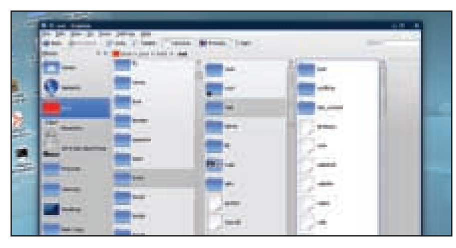
Еще больше путаницы вызывают виртуальные файлы и каталоги. Один из таких каталогов – **/proc**. С виду это обычный каталог, такой же, как любой другой. Но, взглянув на его содержимое, вы обнаружите экзотическую смесь из чисел, директорий и символических ссылок. При попытке определить, кто является владельцем файлов, через графический файловый менеджер или по команде **ls -l /proc**, вы увидите множество имен, в том числе системного администратора (root), ваше собственное и кучу других, включая демонов и фоновые задачи.

Причина тут в том, что каталог **/proc** – виртуальная файловая система. Все его файлы и папки на самом деле не хранятся на вашем жестком диске. Они созданы ядром, чтобы пользователи и приложения могли получать сведения обо всем изобилии процессов, аналогично доступу к файлам. Например, введя команду **cat /proc/meminfo**, вы увидите разнообразную информацию о конфигурации вашей памяти, включая объем памяти, свободной на данный момент. Команда **cat /proc/cpuinfo** покажет тип процессора, установленного на вашем компьютере. Числа, которые вы видите в этом каталоге – это идентификаторы всех работающих процессов для каждой из запущенных вами задач. Те же номера вы увидите при просмотре списка задач с помощью графического монитора системы или по команде **top**.

Мало того...

В каталоге **/proc** найдется любая информация о любой задаче. Например по команде **cat /proc/1/cmdline** вы увидите команду, запустившую первый процесс, **/sbin/init**.

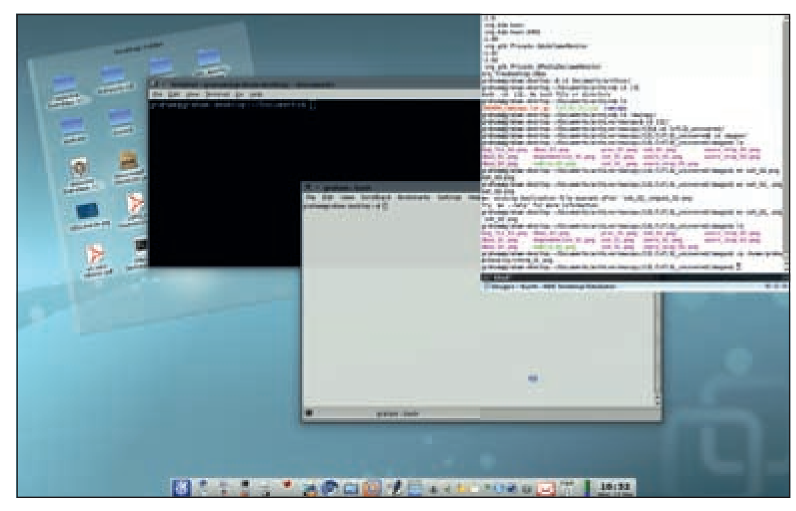
Если вам нужна более общая информация, займитесь еще одной виртуальной файловой системой: **/sys**. Как и его эквивалент, относящийся к процессам, этот каталог полон виртуальных файлов и папок с данными о вашей системе. В него вложены директории **block**, **bus**, **class**, **dev**, **devices**, **firmware**, **fs**, **kernel**, **module** и **power**; каждая из них соответствует одному из компонентов системы, играющих принципиальную роль. **block** содержит информацию об устройствах-накопителях (также называемых «блочными»), а **kernel** позволяет увидеть, что именно происходит на самом низком уровне системы. **devices** же обеспечивает доступ к драйверам ядра, управляющих работой всех аппаратных компонентов, подключенных к компьютеру, что подводит к глубинам знаний о том, как управляются устройства и само ядро.



› **Каталог /proc состоит из виртуальных файлов и папок с данными о запущенных процессах.**

стрибутивов установленные модули располагаются в каталоге **/etc/pam.d**. Если вы работаете в Gnome, то существует как минимум один модуль для аутентификации вашего имени пользователя на экране *Gdm*, а также для выполнения автоматического входа в систему.

Имеются общие модули для управления стандартным приглашением к регистрации в командной строке, а также для таких популярных команд, как *passwd*, *cvs* и *sudo*. Все они используют PAM для удостоверения, что вы – тот, за кого себя выдаете, и поскольку модули PAM подключаемые, процедура аутентификации не всегда основана на паролях. Есть модули, настраиваемые на использование биометрических данных, например, отпечатка пальца, или аппаратного ключа шифрования (на базе USB-брелка). Самое замечательное в модулях PAM то, что их методы не связаны с тем, откуда вы аутентифицируетесь, а это значит, что можно свободно конфигурировать систему по своему вкусу.



› **Большинство установок Linux предлагают массу способов доступа к терминалу, да еще и не к одному!**

Оболочки командной строки

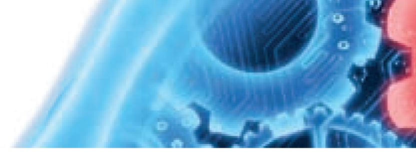
Внутренними механизмами работы вашего компьютера управляют так называемые оболочки [shell]; они могут иметь как графический, так и текстовый интерфейс. До появления графических дисплеев, предоставляющих пользователям интерактивную среду для работы через сеть, нормой были текстовые дисплеи, и этот слой до сих пор остается жизненно важной частью Linux. Текстовые оболочки прячутся под GUI, и часто «выходят на поверхность», когда вам нужно выполнить специфическую задачу, не обеспечиваемую графическим интерфейсом.

Существует масса графических приложений, позволяющих открыть окно в мир командной строки. Приложения *Terminal* (Gnome) и *Konsole* (KDE) – только два из них, самые известные. Однако лучше всего в оболочках командной строки то, что для их использования вообще не нужен графический интерфейс. Возможно, вы уже знакомы с виртуальными консолями. Это приглашения к регистрации в системе, которые появляются, если нажать клавишу Alt и, удерживая ее в этом состоянии, нажать одну из функциональных клавиш (F1–F6). Если вы запустите виртуальную консоль и зарегистрируетесь, указав свое имя пользователя и пароль, в вашем распоряжении окажется полнофункциональный терминал, что особенно удобно, если ваш X-сеанс завершится аварийно и потребуется перезапустить его.

Подобные консоли и сейчас используются многими системными администраторами и обычными пользователями. Виртуальная консоль требует гораздо меньшей ширины канала и легче воссоздается, чем ее графический эквивалент, так как пересылается только текст; это идеальное средство удаленного администрирования. Так что, справившись с обучением, вы сразу же обнаружите, что интерфейс командной строки куда мощнее графического.

По умолчанию, если вы не установите систему *X Window*, большинство дистрибутивов перейдет к оболочке, известной как *Bourne Again Shell* или просто *Bash*. *Bash* – это интерфейс командной строки, который использует большинство из нас. Он позволяет выполнять скрипты и приложения из командной строки, находясь где угодно, в любой точке файловой системы. Если вы не боитесь лаконизма текстового интерфейса командной строки, то с его помощью вы выполните практически любую задачу.

Оболочек очень много, и каждая из них ориентирована на конкретный тип пользователей. Например, можно выбрать интерфейс в стиле программирования на С (*C-Shell*) или сверхмощную оболочку, позволяющую делать что угодно (*Z-Shell*). Однако базовые возможности у них одинаковы, и чтобы извлечь из них максимум, необходимо разобраться в файловой системе Linux. »



Уровень 4 Ядро и его друзья

Именно здесь живет Большой Босс Linux.

Что такое... Переключение контекстов

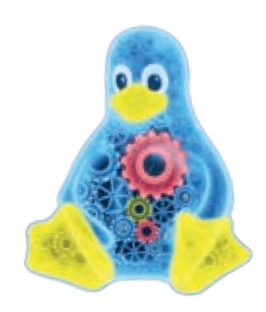
Переключение контекстов – это способность CPU «перескакивать» между выполняемыми процессами и потоками, что создает иллюзию параллельной работы задач.

Мы спускаемся на нижние уровни операционной системы Linux, оставляя позади царство относительной простоты – взаимодействия с пользователем, графических интерфейсов, командных строк… Лучший способ объяснить все происходящее на нижнем уровне – пошагово рассмотреть процесс загрузки, от начала до момента выбора: запустить графический сеанс или начать работать с командной строкой. И в первую очередь (ну, после загрузчика и ядра) вы увидите *init*.

Процесс *init*, используемый большинством дистрибутивов, включая Debian и Fedora, для запуска всех остальных компонентов ОС, должен начать работу в тот момент, когда ядро Linux выполнит инициализацию своих внутренних структур и будет готово создавать пользовательские процессы. Он имеет долгую историю, и версия, используемая большинством дистрибутивов Linux – это *sysvinit*, являющаяся наследием Unix System V.

Запуска требует все – от *Samba* до SSH; процесс *init* выполняет запуски, «прочесывая» каталог со стартовыми скриптами и вызывая их в особом порядке, определяемом номером, с которого начинается имя скрипта. Состав исполняемых скриптов зависит от так называемого уровня работы (runlevel) вашей системы. Их различие определяется используемым дистрибутивом и особенно заметно между потомками Fedora и Debian.

Изучить их в действии можно, используя команду *init* для переключения уровней runlevel вручную. В системах на базе Debian, команда **init 1** переводит систему в однопользовательский режим, а команда **init 5** – в полномасштабный графический режим. А старые версии Fedora предлагают консоль без сетевых возможностей (runlevel 2), консоль с сетевыми возможностями (runlevel 3) и полномасштабный графический режим (runlevel 5), причем каждый процесс будет запущен в свою очередь, по мере загрузки системы. Это может создать узкое место, особенно когда один процесс ждет активизации сетевых сервисов. Каждый скрипт дожидается завершения предыдущего, и только после этого может начать работу, независимо от того, какие объемы системных ресурсов остаются в простое.



Если вы думаете, что система *init* здорово устарела, вы не одиноки. Так считают многие, и разработчики ряда дистрибутивов уже рассматривают переход с *init* на альтернативный вариант, *upstart*. Что следует упомянуть особо, дистрибутив, который в настоящее время спонсирует разработку *upstart*, Ubuntu, уже использует его по умолчанию в качестве загрузочного демона. То же самое делает Fedora, а разработчики Debian тоже объявили о намерении использовать *upstart* в следующем релизе своего дистрибутива.

Серьезным преимуществом *upstart* является асинхронный запуск скриптов. Это означает, что пока один из них ожидает появления сетевого соединения, другой в это время может настраивать аппаратную часть или инициировать *X*. *Upstart* будет использовать те же скрипты, что и *init*, но сделает процесс загрузки более быстрым и эффективным. Это – одна из основных причин, по которым последние версии Ubuntu и Fedora по сравнению со своими предшественницами загружаются намного быстрее.

Ядро

Итак, мы рассмотрели почти все, с одним, но очень крупным, исключением – самим ядром. Как уже говорилось, ядро отвечает за поддержку всех системных ресурсов и управление ими. Это «сердце» работающей системы Linux; именно оно делает Linux тем, что он есть. Ядро управляет файловыми системами и процессами, загружает драйверы, реализует сетевые средства, пользовательские пространства, управляет памятью и дисками. И, как ни странно, обычному пользователю здесь практически не на что смотреть. Кроме элементов, отображаемых при просмотре виртуальных файловых систем **/proc** и **/sys**, а также ряда процессов, работающих в фоновом режиме, большинство управляющих систем абсолютно прозрачны.

> «Команда lsmod выдаст список модулей, подключенных к ядру.»

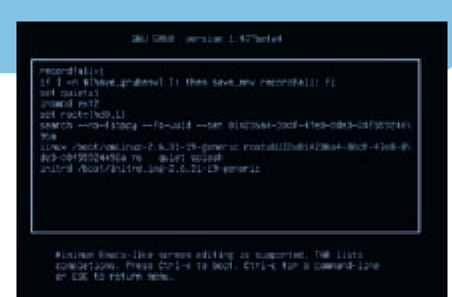
Но есть и видимые элементы, и самый заметный – инфраструктура драйверов, управляющих аппаратными средствами. Разработчики большинства дистрибутивов предпочитают паковать драйверы как модули, а не как часть монолитного ядра, и это значит, что драйверы могут загружаться и выгружаться по мере на-

Grub

Grub появляется первым, еще до того, как начинает загружаться Linux. Обычно он запускается из главной загрузочной записи (MBR) вашего первого жесткого диска при старте компьютера. Кроме Linux, *Grub* может загружать Windows и Mac OS X, и обычно позволяет выбирать операционную систему из загрузочного меню. Как правило, конфигурация для каждой операционной системы хранится в загрузочном каталоге на разделе Linux. Именно оттуда вы можете изменять загрузочные параметры, доступные через меню, хотя большинство дистрибутивов предлагают и графический инструмент для упрощения этой процедуры.

Запись для Linux, например, указывает на двоичный образ ядра, а также на диск, где он находится. Это одно из отличий *Grub* от его предшественника, *Lilo*: *Grub* может читать разделы Linux на диске и загружать любой из найденных образов ядра. Благодаря этой возможности, загрузочное меню *Grub* интерактивно. Здесь можно менять различные загрузочные параметры, выбирать ядро для загрузки и раздел жесткого диска, с которого требуется выполнить старт системы. Для этого выделите в загрузочном меню нужную строку, нажмите клавишу E и отредактируйте текст с помощью встроенного редактора. В мультизагрузочной системе, не только с Linux, но и с другими ОС, возможность настройки параметров может оказаться настоящим спасением.

После выбора Linux из меню загружается ядро Linux, которое берет на себя управление дальнейшим процессом. Именно на этой стадии на экране начинают появляться различные сообщения о ходе загрузки (если в вашем дистрибутиве они не скрыты). RAM-диск (**initrd**) копируется загрузчиком *Grub* в определенную область памяти и затем используется ядром как временная файловая система Linux, содержащая, например, драйверы



› По нажатию клавиши E, в загрузочном меню можно редактировать опции загрузки *Grub* на лету, не фиксируя изменений.

жесткого диска и файловой системы для монтирования реального корневого раздела. После этого управление передается расположенному в нем «настоящему» процессу *init*.

добности. Какие модули ядра включены, а какие нет, зависит от вашего дистрибутива. Но если у вас установлен исходный код ядра, обычно можно без особого труда собрать собственные модули; или установите их через менеджер пакетов вашего дистрибутива.

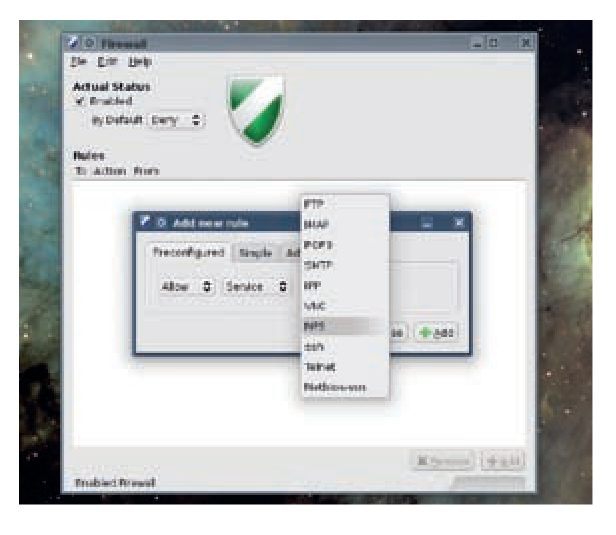
Чтобы увидеть, какие модули подгружены к ядру в данный момент, введите команду **lsmod** от имени системного администратора. Рядом с именем каждого модуля помещается список зависимостей – это другие программные компоненты, необходимые для корректной работы модуля.

Модули зависят от ядра, и именно из-за этого ваш драйвер Nvidia может внезапно отказать, если ваш дистрибутив автоматически обновит ядро. Сборка модуля Nvidia GLX выполняется для текущей версии ядра, что и происходит при запуске инсталлятора. К счастью, можно установить несколько версий модуля, и каждая из них при выборе нового ядра из загрузочного меню *Grub* будет обнаруживаться автоматически – потому что все различные модули спрятаны в каталоге **/lib/modules**, который, в свою очередь, содержит дальнейшие каталоги, названные в соответствии с версиями ядра. Чтобы определить, какую версию ядра используете вы, скомандуйте **uname -a**.

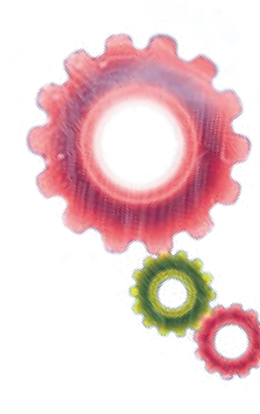
В зависимости от вашего дистрибутива, вы можете найти множество модулей драйверов ядра в каталоге **/lib/modules/версия_ядра/kernel/drivers**. Просмотр этого каталога бывает полезен, если ваше оборудование распознается неправильно. Точно зная, какой модуль должно использовать аппаратное устройство, вы можете попробовать загрузить его командой **modprobe** с именем нужного модуля. Тогда может оказаться, что ваше устройство работает и не нуждается в дальнейшей настройке. Тем не менее, благоразумно будет просмотреть системные журналы и убедиться, что ваши аппаратные средства работают как полагается. Удалять модули из памяти можно с помощью команды **rmmod**; это пригодится в случаях, когда установщик драйвера Nvidia сообщает, что драйвер уже работает.

Iptables

В списке, выводимом командой **lsmod**, присутствует необычный модуль – **ip_tables**. Он является частью одной из мощнейших подсистем Linux, сетевой безопасности, и ядро использует его для реализации брандмауэра Linux. Брандмауэр применяет сложную систему правил для управления всеми сетевыми пакетами, поступающими на компьютер и покидающими его. Командой *iptables* можно изменять конфигурацию брандмауэра в режиме реального времени. Но если вы не эксперт в данной области, эта задача может оказаться сложной для понимания, особенно если ваш компьютер подвергается риску вторжения. Это является отражением сложности сетевого стека, а не модуля *Iptables* как такового, и представляет собой неизбежный побочный эффект попыток управлять различными уровнями сетевых данных одновременно.



› Не хотите возиться с ручной настройкой *Iptables* – и не надо: куча клиентских приложений с GUI, например, *GUFW*, упростят для вас эту задачу.

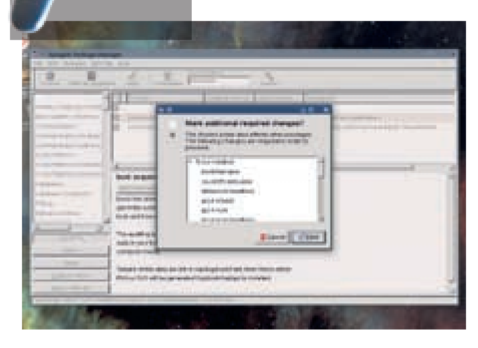


Но, если вы привыкли к другим системам и хотите конфигурировать *Iptables* вручную, рекомендуем воспользоваться приложениями с графическим пользовательским интерфейсом, например, *Firestarter* или входящей в состав Ubuntu программой *ufw*. Они созданы специально для того, чтобы упростить настройку *Iptables*. Установив *ufw*, вы можете быстро активизировать брандмауэр, войдя в систему как root и введя команду **ufw enable**. Можно разрешать или блокировать конкретные порты с помощью команд **ufw allow** и **ufw deny**, или заменить порт именем сервиса, который вы хотите блокировать. Список сервисов, работающих в системе, находится в файле **/etc/services**, и при полной решимости вы можете использовать еще более дружелюбное клиентское приложение для настройки *Iptables*, установив пакет **gufw**.

Это еще не все...

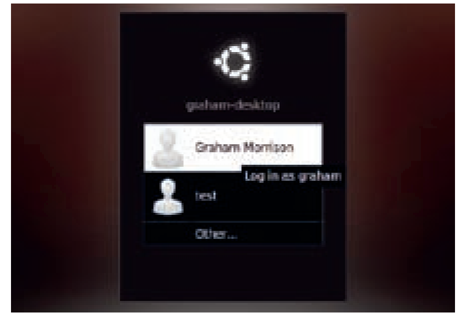
Мы рассмотрели важнейшие аспекты операционной системы Linux, и надеемся, что теперь вы гораздо лучше понимаете, как взаимодействуют все эти компоненты. Одной из лучших черт Linux является свобода эксперимента: менять можно все. И это лучший способ изучить операционную систему и ее возможности – конечно, если вы не делаете этого в производственной обстановке. Тогда попробуйте запустить любимый дистрибутив в виртуальной среде, а если вам нужны помощь и пояснения, загляните на форумы **LXF** по адресу: www.linuxformat.ru/forum. LXF

Шаг за шагом: Создаем диаграмму загрузки



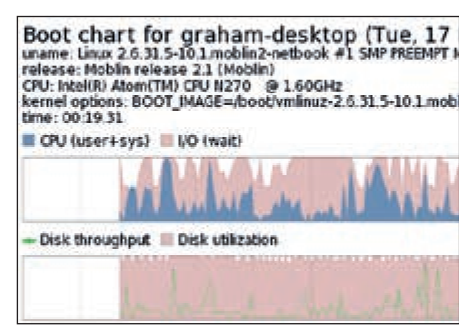
1 Установим пакет

Воспользуйтесь своим менеджером пакетов для установки **bootchart**. Это модифицирует файл настройки загрузчика *Grub*.



2 Перезагрузим компьютер

Перезагрузите компьютер. В процессе загрузки процесс *bootchart* запустится одним из первых, и будет вести журнал всех событий.



3 Просмотрим диаграмму

Перейдите в каталог **/var/log/bootchart** и поищите там PNG-файл. В нем отображается точная информация о том, что именно загружается и когда.

Летать выше

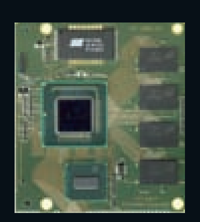
Николас Рой объясняет **Антону Борисову**, как исследовать слишком опасные, холодные или далекие для человека места.



Беспилотные летательные аппараты или БЛА [Unmanned Aerial Vehicles, UAVs] становятся все умнее и выполняют все более сложные задачи. Они помогают исследовать удаленные или опасные места. Николас Рой [Nicolas Roy] из группы Robust Robotics Group Массачусетсского технологического института [MIT] — глава команды, выигравшей в прошлом году международный конкурс по робототехнике со своим квадрокоптером. И представьте себе, квадрокоптер управляется Linux…

Linux Format: Расскажите, пожалуйста, о ваших БЛА. Откуда взялась эта идея?

Николас Рой: Мы занялись ею года три назад. Руйджи Хе [Ruijie He], который тогда был студентом-дипломником, вместе со мной начал работать над сумасшедшей идеей создания вертолета с лазерным дальномером, способного пролететь в открытую дверь. Это был его дипломный проект, и мы по ходу дела допустили массу конструкторских ошибок – весьма и весьма поучительных. Следующим летом мы обратились в компанию Ascending Technologies, которая делает для нас летательные аппараты (ЛА). Их ЛА стабильны, надежны и удобны в работе, так что контракт с ними был огромным шагом вперед. Вскоре к нашему проекту присоединились Эбрахам Бахрах [Abraham Bachrach], Сэм Прентис [Sam Prentice] и Гарретт Хеманн [Garrett Hemann] – команда набрала «критическую массу», и проект стал быстро прогрессировать.



LXF: Квадрокоптер гораздо сложнее вертолета или самолета. Есть ли для них ниша на рынке?

НР: Говорить, что квадрокоптеры «намного сложнее», неверно. У каждой конструктивной схемы ЛА есть свои достоинства и недостатки – и у самолетов с неизменяемой геометрией крыла, и у традиционных винтокрылых ЛА, и у квадрокоптеров. Основная причина, по которой мы решили использовать вертолет – его умение «зависать» над объектом, недоступное обычному самолету [это требует изменяемой геометрии крыла и способности вертикального взлета и посадки, – *прим. пер.*]. Сложности управления квадрокоптером по сравнению с традиционным вертолетом зависят не столько от ЛА, сколько от выполняемых им функций.

LXF: И все же — почему вы остановились на четырехмоторной схеме, а не на одномоторной?

НР: Конструкция квадрокоптера, предложенная Ascending Technologies, проще, стабильнее и управляемее, чем традиционная. Повышение стабильности, естественно, ограничивает динамику аппарата, хотя при текущих допусках наших датчиков мы, вероятно, пока и не сможем восстановить контроль над квадрокоптером после слишком стремительных маневров. Но мы работаем над этим.

LXF: На чем летают ваши БЛА?

НР: Наши ЛА и их двигатели разработаны в Германии компанией Ascending Technologies. Заказная аппаратура, которую они для нас изготовляют, великолепна, и мы доверяем им на все 100 %. У ЛА четыре электродвигателя, по штуке





О РЕЛИЗЕ ПОД GPL

«Добившись стабильности ПО, мы, видимо, выпустим релиз.»



на ротор. Варьируя скорость вращения двигателей, можно менять тангаж или крен квадрокоптера. Управление высотой ЛА, в свою очередь, создает механический импульс. ПК с процессором Atom 1,6 ГГц с картой SD взаимодействует через последовательный интерфейс с микроконтроллером на базе ARM. Кроме того, имеется Wi-Fi-связь с ПК на базе Atom.

LXF: Почему вы выбрали Linux?

HP: Во-первых, со временем у нас накопилась обширная кодовая база. Код, управляющий нашим вертолетом, был разработан на базе пакета для управления наземным роботом, он доступен здесь: http://carmen.sourceforge.net. Открытая природа Linux обеспечивает «прозрачную» работу с оборудованием типа бортового микроконтроллера и датчиков. При неполадках в управлении оборудованием приятно иметь возможность прошагать через все драйверы и понять, что происходит.

При всем при том, в выборе инструментария мы прагматичны, и Linux – отнюдь не «серебряная пуля» за все. В нашей лаборатории применяются не только различные дистрибутивы Linux, но и Windows, и Mac OS X, в зависимости от того, что мы делаем.

LXF: Вы пробовали использовать для управления БЛА ядро реального времени?

HP: Нам не понадобилось ядро реального времени, потому что циклы самого низкого уровня, управляющие двигателями, берет на себя микропроцессор от Ascending Technologies. Конечно, сигналы поступают от датчиков с некоторым «дрожанием» [jitter], которое ядро реального времени могло бы исправить, но эта помеха относительно мала.

LXF: Используете ли вы другие открытые библиотеки, например, *OpenCV*, или полностью полагаетесь на собственные разработки?

HP: Да, конечно, мы используем *OpenCV*, а также библиотеки примитивов производительности от Intel и другие пакеты. Если есть библиотека, дающая нужные нам функции – мы ее и берем.

LXF: А почему вы не использовали GPS? И каковы потенциальные области применения вашего прототипа?

HP: Чтобы ориентироваться с помощью GPS, обычно нужно видеть три или четыре спутника. Поэтому GPS не будет работать, например, в гараже. Кроме того, GPS очень ненадежно работает в городах, так как спутниковые сигналы ослабляются при отражении от зданий.

В итоге большинство БЛА летают на минимуме высоты. А мы исследуем полеты в среде без GPS. Мы хотим летать повсюду. Легко представить, как наши аппараты помогают в проведении аварийно-спасательных работ, поиске пострадавших в зонах бедствия или определении, не опасно ли войти в здание. Есть и применения в военных целях.

LXF: Предлагаете ли вы (или планируете предложить) свое ПО под GPL или какой-либо другой «свободной» лицензией?

HP: Наш аппарат – готовый продукт Ascending Technologies, поэтому он уже доступен. Наше ПО разработано на основе пакета *Carmen* для управления наземным роботом, свободного под GPL. Добившись стабильности ПО для управления вертолетом, чтобы оно было пригодно для любой сторонней группы, мы, видимо, выпустим релиз.

LXF: А сработался бы ваш квадрокоптер с «марсоходами»?

HP: Мы уже создали подобный прототип, в сотрудничестве с Джоном Леонардом [John Leonard] и Сетом Теллером [Seth Teller] из MIT – для смешанной группы наземных роботов и БЛА, с целью исследования незнакомой местности. Куда-то не добраться вертолетам, а куда-то – наземным роботам. Однако Марс как область для полетов мы не рассматривали. LXF

Cygwin и MSYS

Добро пожаловать из тьмы к свету

О пользователях Windows важно помнить, что это не их вина. Вливайтесь в миссию **Марка Бэйна**, чтобы вывести их к свету – расскажите им о *Cygwin* и *MSYS*.

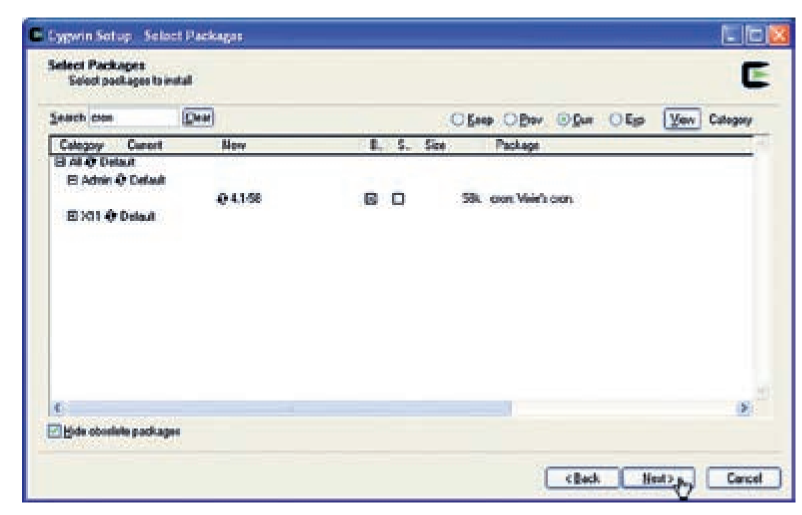


Как достойные провозвестники свободного ПО, мы всегда восхваляем преимущества Linux перед проприетарными ОС. Но иногда полный переход в Linux-окружение бывает невозможен. К примеру, на раздаче в столовой Звезды Смерти могут быть только компьютеры с Windows, и Империя не хочет (или не может) перейти на Linux, потому что зависит от приложений вроде *Microsoft Access*. И как же вам продолжать использовать Windows (по обязанности) и в тоже время применять функциональность Linux (типа сценариев оболочки)?

Дело в том, что существует пара прекрасных Linux-подобных окружений, которые вы можете установить в Windows: *MSYS* (Minimal System) и *Cygwin* (GNU + Cygnus + Windows). А как это может помочь вам в вашей ежедневной работе на Windows-машине? Давайте взглянем на *MSYS*.

Вы никоим образом не должны допускать ошибку, считая *MSYS* полноценным дистрибутивом Linux. Это просто *Bourne Shell*, перенесенный под Windows, что стало возможно благодаря порту *GCC* – *MinGW* (Minimalist GNU for Windows). На секундочку: как вам идея – *Bourne Shell* для Windows?!

Установка очень проста: скачайте и установите текущую версию *MinGW* с http://sourceforge.net/projects/mingw/files/Automated_MinGW_Installer, затем скачайте и установите *MSYS* с http://sourceforge.net/projects/_mingw/files/MSYS_Base_System/msys-1.0.11/MSYS-_1.0.11.exe/download.



› *Cygwin* умеет многое из того, чего не хватает в *MSYS*: например, добавлять пакеты во время установки.

MSYS

Ну вот, вы установили *MSYS* — и что с ним теперь делать? А то же самое, что и с *Bourne Shell* на Linux-машине. В частности, вас особо впечатлит полное отсутствие в нем хоть какого-то намека на графический интерфейс. Здесь все делается из командной строки, без каких-либо скрытых опций и дополнений.

Да, *MSYS* – просто голая оболочка. Посмотрим, к примеру, много ли файлов и папок в **My Documents**. Для этого запустите ваш новый *Bourne Shell*, одним из трех способов:

» Выберите *MSYS* в меню Windows

» Два раза щелкните мышью на **msys.bat** в **C:\MSYS\1.0**.

» Вызовите *MSYS* из **C:\msys\1.0\msys** в командной строке Windows.

Затем начинайте печатать, как будто вы находитесь в Linux (вам будет приятно узнать, что при этом сохраняется функция автодополнения по двойному нажатию клавиши Escape):

```
ls -l /c/Documents\ and\ Settings/Mark/My\ Documents/l wc -l
```

И, конечно же, можно использовать *awk* (или *gawk*) для обработки файлов наподобие **expenses.csv**:

```
еда,25
```

```
бензин,50
еда,25
бензин,20
```

Замечательно то, что создавать файл сценария вы можете и в своем любимом Windows-редакторе (типа *Notepad++*):

```
EXPENSES="/c/Documents and Settings/Mark/My
Documents/expenses.csv"
ITEMS="$(awk -F, '{
print $1
}' "$EXPENSES" | sort -u)"
for ITEM in $ITEMS
do
awk -F, -v ITEM=$ITEM '
BEGIN {TOTAL=0}
{if ($1==ITEM) {TOTAL+=$2}}
END {print ITEM,TOTAL"руб"}
' "$EXPENSES"
done
```

Его надо сохранить в **C:\msys\1.0\msys\home\<ваша_учетная_запись>\expenses**. При новом старте *MSYS* вы сможете сделать

```
chmod +x expenses
./expenses
```

Разумеется, в этом месте вам захочется применить такой инструмент Linux, как *Cron*, но тут-то вы и обнаружите полную минималистичность *MSYS*. Это, как мы говорили, абсолютно голый Linux. Однако идея, стоящая за *MinGW*, заключается в возможности портирования ваших Linux-инструментов на Windows, и вы найдете ряд таких инструментов (к примеру, *Emacs* и *wget*) в составе проекта MingwPORT на http://sourceforge.net/projects/mingw/files/mingwPORT/Current Releases.

Cygwin

Допустим, вы все же сочли, что *MSYS* для вас скудноват. Тогда, возможно, вы решите ознакомиться с *Cygwin*. Перед этим, однако, стоит повторить, что *Cygwin* – Linux-подобное окружение, а не дистрибутив Linux, собранный для работы с Windows. Он фактически состоит из DLL (динамически подгружаемой библиотеки) Windows и набора инструментов для работы с ней. И это важно помнить. Нельзя взять любое Linux-приложение и заставить его работать с помощью *Cygwin*. Тем не менее, инструментов достаточно, чтобы ваши труды окупились.

Вы можете скачать инсталятор *Cygwin* с www.cygwin.org/cygwin. Загрузите его и запомните, куда вы это сделали – позже нам это понадобится. Сохранив **setup.exe**, запустите его и следуйте инструкциям.

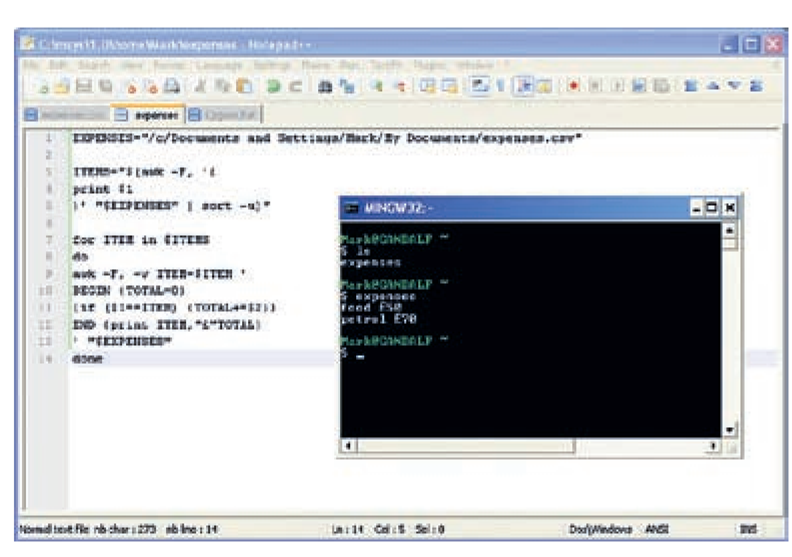
Для вас создастся базовая оболочка *Bash*, к которой можно обратиться одним из трех способов:

» Дважды щелкнуть по иконке на рабочем столе.
» Выбрать Cygwin Bash Shell в меню Windows.
» Запустить **C:\cygwin\Cygwin.bat**.

Cygwin.bat любезно подскажет вам, как запустить *Cygwin* из командной строки Windows:

```
chdir C:\cygwin\bin
bash --login -i
```

Независимо от того, как вы намерены применять *Cygwin*, вы обнаружите, что это очень похоже на использование *MSYS* (вы также получаете доступ к консоли и набору команд Linux); зато базовый набор команд в *Cygwin* больше. Возможности *Cygwin* раскрываются в полной мере в том случае, если вам необходимо добавить в систему дополнительную Linux-функциональность – к примеру, задания *Cron*, только ради которых уже стоит задуматься о Linux. Но ни *Cygwin*, ни *MSYS* не включают *Cron* по умолчанию, и если вы скомандуете



› ***MSYS*** **придаст мощь *Bourne Shell* (и все его преимущества при работе с файлами) вашему компьютеру с Windows. Почувствуй силу!**

```
crontab -l
```

оба вернут вам ошибку. Вот тут снова пригодится **setup.exe**. Проберитесь через экран выбора пакетов и, применив поиск, найдите пакет *Cron*. Вам также понадобится установить *cygrunsrv* (запуск сервисов *Cygwin*) и *Nano* (простой, но эффективный текстовый редактор Linux, про который мы говорим на стр. 38). Покончив с этим, вы сможете запустить демон *Cron*:

```
cygrunsrv -I cron -p /usr/sbin/cron -a -D
net start cron
```

Не исключено, что вернется ошибка, но не волнуйтесь: на самом деле он запустился. Проверьте это, набрав

```
ps -efl grep cron
```

Далее установите *Nano* как редактор для *Cron* –

```
export EDITOR=nano
```

и отредактируйте свой cron-файл:

```
crontab -e
```

Теперь добавьте данные для вашего задания *Cron*:

```
* * * * * echo "$(date)" > ~/test.txt
```

Нажмите Ctrl+X для записи и убедитесь в появлении в вашей домашней директории нового файла – задание *Cron* должно создать и затем ежеминутно обновлять его.

> **«Cygwin состоит из DLL Windows и инструментов для работы с ней.»**

И наконец, чтобы вы не путались, работая сразу с Windows и Linux, *Cygwin* предоставляет прекрасный инструмент для конвертации путей формата Windows в формат Linux (точнее, Unix) и наоборот:

```
cygpath --windows ~ # это должно вернуть (к примеру) C:\
cygwin\home\Mark
cygpath --unix C:/ # это должно вернуть /cygdrive/c
```

Итак, за кратчайший срок мы перешли от Windows-сценариев (фу! Бяка!) к Windows+Linux-сценариям (ура!). Но помните – это не полноценное решение для запуска Linux-приложений под Windows. Правильно будут работать только приложения, скомпилированные с помощью *MinGW* или *Cygwin*. Тем не менее, это не какой-то доморощенный компромисс или бессмысленное упражнение в технике – это достойное решение для желающих перенести Linux-сценарии в Windows. И не давным-давно в далекой-далекой галактике, а сегодня и на Windows-компьютере рядом с вами. **LXF**

ClamAV: Ставим антивирус

В развитие темы безопасности, **Боб Мосс** расскажет, как защитить систему от вирусов и вредоносного ПО.



Наш эксперт

Боб Мосс делит свое время между программированием и обучением на факультете вычислительной техники.

В прошлом месяце мы научились устанавливать и настраивать SELinux так, чтобы снизить ущерб, который вредоносное ПО способно причинить нашей любимой Linux-системе. В этот раз мы шагнем дальше и узнаем, как найти и изолировать вирусы в компьютерной сети, прежде чем прибегать к SELinux или *AppArmor* в надежде, что они спасут положение. Мы также затронем важный вопрос: так ли безопасен Linux, как следует из его репутации?

Пользователи Gnome, вероятно, найдут *ClamAV* под псевдонимом 'Virus Scanner' [Антивирус]: проект *ClamTK* предлагает для программы дружелюбный интерфейс на основе библиотеки *gtk2-perl*. При установке пакета *ClamTk* из репозитория своего дистрибутива вы можете обратить внимание, что эта оболочка не самой последней версии, но, к счастью, пакеты для любой системы на основе Red Hat или Debian можно загрузить с сайта http://clamtk.sourceforge.net, а пользователи других дистрибутивов могут собрать ее из исходников обычным образом. Пользователи KDE также могут обратить внимание на *KlamAV* или его замену *clamav-kde*, доступные в репозиториях большинства дистрибутивов.

> «Мы затронем спорный вопрос: так ли безопасен Linux?»

Однако, хотя эти интерфейсы продуманы и интуитивно понятны, их функционала хватает только на выполнение самых общих задач; и если вы избрали этот путь, оболочки придется запускать вручную. Гораздо больше мощи и гибкости прячется внутри.

ClamAV доступен в большинстве менеджеров пакетов, но версия в вашем дистрибутиве скорее всего не самая последняя, и вам может не достаться новейшего функционала или более высокого уровня безопасности. Пользователи Ubuntu могут просто перейти в System > Administration > Software Sources [Система > Администрирование > Источники ПО] и выбрать вкладку «Third Party Repository» [Сторонние репозитории], чтобы добавить туда PPA [Personal Package Archive – Персональный архив пакетов]. Наши постоянные читатели вспомнят, что мы рассказывали о PPA в Ubuntu еще в **LXF124**, но пользователи Ubuntu Karmic могут воспользоваться следующим удобным сокращением:

```
ppa:ubuntu-clamav/ppa
```

В более старых версиях Ubuntu наберите вот что:

```
deb http://ppa.launchpad.net/ubuntu-clamav/ppa/uebuntu jaunty main
```

Не забудьте заменить **jaunty** соответствующим кодовым именем вашей версии. Закрыв диалог добавления репозитория, откройте терминал и выполните следующую команду, чтобы сообщение об ошибке, которое вы только что видели, больше не появлялось:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 5ADC2037
```

Далее, произведите обычное обновление, чтобы загрузить актуальный список репозиториев и пакетов. Пользователи других дистрибутивов могут скачать архив с последней версией исходного кода *ClamAV* с сайта http://clamav.net/download.

Вскрываем ракушку

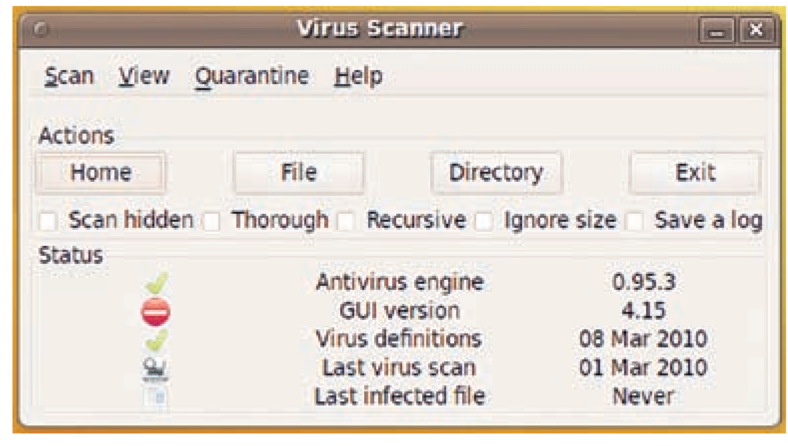
Теперь, когда репозитории или пакеты с исходными текстами готовы, пора установить *ClamAV*. Вы можете выбрать, как он будет запускаться – в виде отдельного приложения или в режиме фонового демона (тогда потребуется добавить пакет *clamav-daemon*). В обоих случаях будет установлен *clamav-freshclam*, который будет поддерживать сигнатуры вирусов (и даже сам *ClamAV*) в актуальном состоянии.

Приложение *ClamAV* запускается из командной строки, вручную или из скрипта, а демон постоянно работает в фоне. И то, и другое можно сделать позже, а сейчас нужно только решить, как вы хотите поддерживать антивирусную защиту в своей Linux-системе.

Установив выбранный пакет, перед первой проверкой компьютера обновите вирусные сигнатуры. Это можно сделать через графический интерфейс или просто командой с терминала:



› **Если *ClamAV* установлен как демон, можно проверить, запущен ли он, всего одной командой.**



› **Для Gnome, *ClamTK* предоставляет дружелюбный и интуитивно понятный интерфейс к *ClamAV*, а пользователям KDE нужно обратиться к *clamav-kde*.**

```
sudo freshclam
```

Постоянные читатели вспомнят, что мы кратко упоминали, как фильтровать входящие пакеты с прокси-сервера с помощью *ClamAV*, в учебнике о родительском контроле в **LXF128**. Настроить *ClamAV* для работы с заданным прокси-сервером проще простого. Достаточно открыть файл **/etc/clamav/freshconf.conf** и добавить в его конец следующие строки:

```
HTTPProxyServer ip-адрес.сервера
HTTPProxyPort номер_порта
```

Замените **ip-адрес.сервера** и **номер_порта** на соответствующие значения. Если вы предпочли выполнение *ClamAV* в виде демона, перезапустите его, чтобы изменения вступили в силу.

Демонизмы

Если вы запускаете *ClamAV* как демон, неплохо убедиться, что он на самом деле работает. Проверяется это командой

```
ps ax | grep clamd
```

В большинстве систем вы увидите три строки вывода. Если демон не запущен, выполните в терминале эту команду:

```
sudo clamd
```

Версию демона можно проверить с помощью команды:

```
clamdscan -V
```

»

Скорая помощь

Чтобы вывести все сигнатуры вирусов, выполните команду

```
sigtool --list-sigs
```

Зачем мне антивирус?

Вы, быть может, удивляетесь: а почему мы говорим об антивирусной защите для Linux? Ведь Сеть пестрит бесчисленным количеством утверждений о безопасности Linux, а если ввести «**Linux antivirus**» в Google, первой в списке будет статья с Linux.com под заголовком «Note for Linux Newcomers: No antivirus needed» [«Новичкам в Linux: антивирус не нужен»]. Кроме того, доля рынка настольных систем у Linux очень мала, а именно она часто является главным мотивом для взломщиков. Все это порождает вполне резонный вопрос: чем проникновение в Linux может пригодиться злоумышленнику?

Вопреки мифам, иммунитет Linux к вирусам велик, но не безмерен. Эксплойты для кроссплатформенных технологий, типа Java и Flash, приложимы к Linux в той же мере, как и к другим операционным системам. А так как Linux проник на промышленный рынок и web-серверы, вторжения становятся гораздо более доходными, чем были раньше. Данные можно украсть и продать «похитителям личностей» или использовать для создания ботнетов и другой противоправной деятельности.

Если в сети есть компьютеры с Windows (или с двойной загрузкой), Linux может стать «тихой гаванью» для вредоносного ПО Windows, которое может расползтись по локальным дискам Windows или размножиться по всей сети. Хотя коммерческие брандмауэры настольных систем отчасти сдерживают эту угрозу, клиентские компьютеры, получающие вредоносные пакеты с доверенного устройства локальной сети, будут в опасности.

Но прежде чем вы спишете со счетов все свои предыдущие мнения о безопасности Linux-окружения, рассмотрим кое-какие цифры.

По оценкам из различных источников, число эксплойтов для Windows в несколько тысяч раз превышает этот показатель у нашей любимой свободной операционной системы, и благодаря открытой природе Linux большинство уязвимостей ликвидируются сразу после их обнаружения.

Безопасность сообщества

Однако причина, по которой Windows обладает большинством уязвимостей, главным образом кроется в ее доминировании на рынке. Но в отличие от Windows вы можете спать спокойно, зная, что если ваш компьютер заразится, то утилиты вроде SELinux (о нем мы говорили в **LXF130**), а также права доступа к файлам для пользователя, группы и прочих ограничат ущерб, который может нанести системе любое вредоносное ПО.

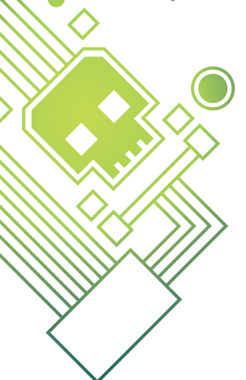
Однако для критически важных систем и бизнес-решений, обязанных удовлетворять федеральным законам о защите данных, есть тысячи свободных средств защиты: от обнаружителей вторжения (IDS), антивирусов и брандмауэров до надежного шифрования данных.

Все это дает более чем достаточную защиту для Linux-системы, и как обычный пользователь, вы получаете гораздо больше преимуществ от разделения этих утилит с сообществом, чем мог бы мечтать любой пользователь Windows. Кроме того, в большинстве случаев вам не нужно платить за годовую подписку или работать с громоздким приложением, мешающим вашей повседневной деятельности.



Автоматическое сканирование

Настроим *ClamAV* на обновление вирусных сигнатур и проверку системы.



Итак, *ClamAV* установлен. Через удобный графический интерфейс можно просмотреть файлы, находящиеся на карантине, запустить обновление сигнатур вирусов и просканировать небольшие каталоги. Но надолго ли вас хватит выполнять это регулярно? Все мы люди, и легко отвлекаемся от ежедневной рутины, оставляя систему уязвимой. Спокойнее будет влезть «под капот» *ClamAV* и настроить его на сканирование и обновление сигнатур вирусов автоматически – тогда система будет сама и вовремя защищаться от любых неприятностей, на которые вы можете иметь несчастье нарваться, бродя по Интернету.

Числовые диапазоны Crontab

Временная шкала	Минуты	Часы	Дни	Месяцы	Дни недели
Диапазон	0–59	0–23	1–31	1–12	0–6

Испробуем простейшие команды в терминале. Если *ClamAV* выполняется как демон, просто замените все вхождения **clamscan** на **clamdscan**, и получите практически тот же функционал. Наша первая команда рекурсивно проверит файлы заданного каталога на наличие любого вируса из базы вирусных сигнатур:

```
clamscan -r /путь/к/каталогу
```

Команду можно улучшить – пусть отображает свой вывод на экране, а по окончании проверки издает сигнал (**bell**):

```
clamscan -r --bell -i /путь/к/каталогу
```

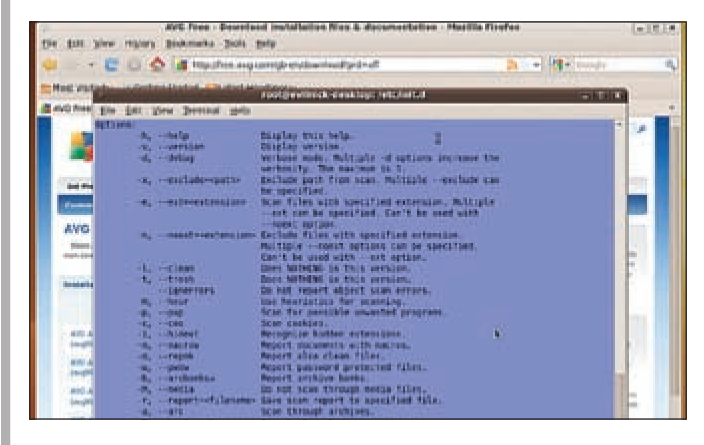
Можно также перенаправить вывод в другие команды или в текстовый файл для последующего просмотра:

```
clamscan -r -i /путь/к/каталогу > results.txt
clamscan -r -i /путь/к/каталогу | mail my@email.com
```

В первом случае результаты рекурсивной проверки помещаются в текстовый файл, во втором – передаются почтовой программе для последующей отправки на указанный адрес. Последний вариант особенно удобен, когда *ClamAV* выполняется в режиме демона: ведь результаты вы будете получать в реальном времени. Наконец, можно добавить перед указанием пути ключ **--remove**, чтобы зараженные файлы автоматически удалялись

Другие антивирусные продукты

Хотя *ClamAV* прост в установке, настройке и управлении, вы можете предпочесть ему готовое решение для защиты настольной системы от вирусов Linux и Windows. Существует несколько коммерческих альтернатив *ClamAV*. Они нацелены в основном на поиск вирусов Windows, нашедших пристанище в Linux, но весьма пригодятся при наличии двойной загрузки или компьютеров с Windows в локальной сети.



AVG Free Antivirus

Сайт: http://free.avg.com/gb-en/download?prd=all

AVG хорошо известен в Windows, а теперь он доступен и в Linux. Интерфейс, по сути, такой же, как в Windows-версии, и предоставляет те же возможности по сканированию. Также есть две платных редакции, предоставляющих дополнительные функции.

Kaspersky

Сайт: http://kaspersky.com/linux

Вы обнаружите, что *Kaspersky* предлагает не только стандартный настольный антивирус, но и полноценные продукты для различных типов серверов, в зависимости от того, какой пакет вы приобретаете. Среди них нет бесплатных продуктов, потому что *Kaspersky* в основном ориентирован на решения для предприятий, но он может подойти для домашних серверов.

Panda Antivirus for Linux

Сайт: www.pandasoftware.com/download/linux/linux.asp

Эта скрытая жемчужина предназначена для запуска на серверах Linux с целью поиска уязвимостей DOS/Windows, но не хуже работает и в системах с двойной загрузкой, и гарантирует, что вредоносные программы для Windows не копируют себя на разделы Windows из Linux. Продукт менее продвинутый, чем другие упомянутые здесь, но свою задачу это легкое приложение выполняет отлично.

Avast Linux Home Edition

Сайт: www.avast.com/linux-home-edition

У *Avast* те же функции, что и у *AVG*; его ключевое отличие – возможность управлять сканированием из командной строки, как у *ClamAV*, и его можно вызывать из скриптов оболочки. Интерфейс основан на родных библиотеках *GTK*, и среди портированных программ эта более всех чувствует себя в Linux, как дома.

из системы, вместо помещения в карантин. Но будьте с этим осторожнее, так как в случае ложных срабатываний можно потерять нужные файлы.

Включить проверку *ClamAV* в системный планировщик очень просто. Просто введите

```
at 3.00 tomorrow
at>clamscan -i ~ > ~/test.txt
```

Отправьте задание, нажав Ctrl+D (**at>** во второй строке набирать не нужно). Команда запланирует антивирусное сканирование на 3 часа утра завтрашнего дня и сохранит результат в файле **test.txt** в домашнем каталоге. А если вам требуется ежедневное сканирование?

Его можно выполнять по-разному. Если вы пользователь настольной системы, запускайте сценарий оболочки при каждом сеансе работы или настройте его вызов в планировщике.

В первом случае просто скопируйте вторую строку предыдущего примера и вставьте ее как новое приложение автозапуска через System > Preferences > Startup Applications (Система > Настройки > Автозапуск). Пользователям KDE нужно сохранить эту строку в скрипте оболочки и скопировать его в каталог **/home/user/.kde/AutoStart**, а затем назначить ему права на выполнение.

Регулярные проверки

Во втором случае можно настроить периодические проверки с помощью *Cron*. Вам потребуется скопировать строку из предыдущего фрагмента кода в файл и назвать его **scanscript.sh**. Предварительно удостоверьтесь, что инструкции в файлах **cron.allow** или **cron.deny** обеспечивают вам доступ к *Cron*. Если ни одного из файлов нет, доступ будет только у суперпользователя-root, но если оба существуют, убедитесь, что ваше имя пользователя есть в первом файле, но не в последнем.

Прежде чем добавить запись в *crontab*, установите свой любимый текстовый редактор в переменной **$EDITOR**. Иными словами, наберите:

```
export EDITOR=nano
crontab -e
```

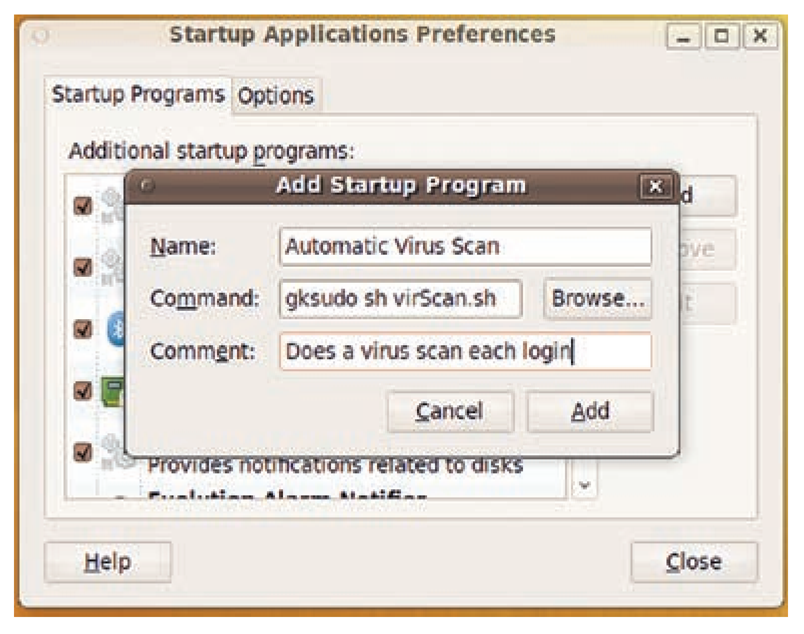
Тогда вы сможете дописать в конец файла такой код:

```
0 * * * * sh /путь/к/scanscript.sh
```

Если в качестве редактора вы выбрали *Nano*, нажмите Ctrl+X для выхода с сохранением. Теперь *Cron* будет запускать антивирус из файла скрипта по часам, каждый час ежедневно. Чтобы понять, как работает эта строка, взгляните на таблицу на соседней странице. День с номером **0** с точки зрения *Cron* – воскресенье. Звездочка * означает, что скрипт будет срабатывать при каждой отметке данной временной шкалы. Справляясь с таблицей, можно задать любой нужный интервал запуска.

> «Обновление сигнатур вирусов следует выполнять раз в день.»

Наконец, если вы не хотите получать от системы письма с результатами выполнения задания, измените предыдущую строку таким образом:



› **Если сканирование в заданное время вам не подходит, выполняйте его в фоновом режиме при каждом входе в систему, создав новую программу автозапуска.**

```
0 * * * * sh /путь/к/scanscript.sh > /dev/null 2>&1
```

Планирование обновлений сигнатур вирусов мало чем отличается от планирования запуска проверки. Однако над периодичностью обновлений нужно немного подумать. Антивирусные проверки лучше запускать почаще, для обеспечения чистоты, но обновления сигнатур обычно производятся не чаще раза в день. Поэтому выполнять их при каждом входе в систему – лишняя трата системных ресурсов.

Свежие сигнатуры

На современных настольных системах накладные расходы не чрезмерны, но их стоит учесть, когда дело касается постоянно включенных сетевых устройств с невысокой производительностью или файловых серверов. На рабочем столе можно запускать обновление вручную через графический интерфейс *ClamTK*, но всегда есть опасность об этом позабыть.

Поэтому советуем выполнять обновление сигнатур вирусов как минимум раз в день, чтобы гарантировать защиту от самых свежих угроз без лишних действий. Наилучший вариант решения этой задачи – добавить в *crontab* следующую строку:

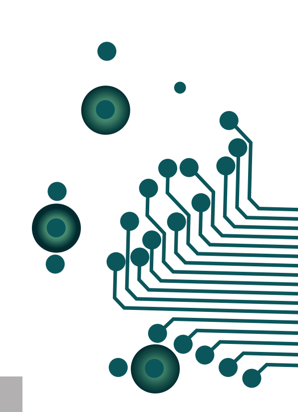
```
0 3 * * * sh /путь/к/scanscript.sh
```

Она гарантирует, что обновления вирусов будут происходить каждый день или, во всяком случае, не в то время, когда потребление системных ресурсов резко возрастает. LXF

Скорая помощь

Если вы получаете сообщения об ошибках «Слишком большой архив» (Oversized Zip), измените параметр **ArchiveMaxCompressionRatio** в файле **/etc/clamd.conf**, предварительно попробовав пару-тройку значений командой

```
clamscan
--max-ratio=400
example.zip
```





› **Обновление сигнатур вирусов можно выполнять вручную; или запланируйте их, так же, как обычные проверки.**

Куда идти дальше?

Мы производили Сравнение различных антивирусов четыре номера назад – в **LXF128**. Материалы на тему безопасности публиковались и в других выпусках журнала: воспользуйтесь списком статей, который находится в каталоге **Magazine/Articles_index** на каждом **LXF**DVD, и поищите по слову «Безопасность». Вы также можете почерпнуть много дополнительной информации на wiki.linuxformat.ru.



Nano: ТРЕТИЙ ПУТЬ

Ненавидите лапидарность *Vi* или комбинации клавиш *Emacs*? **Майк Сондерс** представляет Nano, текстовый редактор, вместивший большую мощь в малый объем.

Похоже, наш мир все более поляризуется. Вам нравится либо Microsoft, либо Apple. Вы пьете либо Coca, либо Pepsi. Вы либо с нами, либо против нас. Время – деньги, и в этом мире стремительных темпов кто может позволить себе выбиться из колеи упорно навязываемого простейшего выбора?

Но вы – пользователь Linux, а значит, избрали третий путь: дорогу, отдельную от двух главных коммерческих систем, Windows и OS X. Вы знаете, что почти всегда найдется вариант, отличный от A и B. Вы сознательно решили попробовать нечто иное, и вознаграждены, помимо прочего, операционной системой, которая стоит сейчас на вашем надежном компьютере. А как насчет текстовых редакторов? Если взглянуть на практически любую дискуссию на эту тему в Интернете, то создастся впечатление, что игру делают всего два редактора: *Vi(m)* и *Emacs*. Оба они чрезвычайно мощные и сложные – между прочим, про *Emacs* часто шутят что это, по сути, операционная система, по случаю прихватившая и текстовый редактор. У них крутая кривая обучения и функциональность, скрытая за тайными командами и вывихивающими пальцы сочетаниями клавиш.

> «Nano содержит массу продвинутых функций, не хуже, чем у Emacs.»

А не рассмотреть ли третий вариант, *Nano* (www.nanoeditor.org)? Этот редактор прекрасен, потому что он

» компактен, легок и включен в большинство дистрибутивов;
» прост в использовании и дает на экран много обратной связи;
» содержит массу продвинутых функций, способных конкурировать с *Vi(m)* и *Emacs*.

Nano работает в текстовом режиме в командной строке, что может навести на мысль «Зачем мне изучать редактор с текстовым режимом? Все, что нужно, делают *Kate, Gedit* и *FooEdit 2000*». Ну, во-первых, все постоянные пользователи Linux и администраторы должны иметь прочный опыт работы с редактором в командной строке. Например, если что-то случится с системой *X Window System*, вы попадете в терминал – и эти навыки окажутся существенными для редактирования конфигурационных файлов.

Во-вторых, если ваша Linux-машина работает как сервер, она не обязательно содержит *X*'ы и графические утилиты; тогда редактор с командной строкой придется применять довольно часто. Базовые навыки работы в нем, помимо открытия и сохранения файлов, увеличат производительность вашего труда, а к этому мы все и стремимся.

Историческая сводка такова: в конце 80-х почтовым клиентом для Unix был *Elm*. Затем возник *Pine*, имя которого, по одной из версий, расшифровывалось как «Pine Is Not Elm (*Pine* – это не *Elm*)». Текстовым редактором в *Pine* был *Pico*, но он не являлся полностью свободной программой в понимании GNU. Поэтому создали *Tip* (Tip Isn't Pico), позднее переименованный в *Nano* (*Nano's ANOther editor*). Забавно, правда?

Прокладываем путь

Прежде чем нырнуть в разбор возможностей *Nano*, уделим минутку на ознакомление с интерфейсом редактора. Ввод **nano** в командной строке выдаст вам *Nano* в состоянии по умолчанию. Если вы уже пробовали *Emacs*, но были огорчены недостатком информации на экране, то *Nano* приятно вас удивит, а уж о *Vi* лучше и не говорить. Можете сразу же вводить текст, используя для перемещения по нему курсорные клавиши.

Вверху находится панель заголовка, содержащая номер версии *Nano* и имя редактируемого файла. Ниже, поскольку мы еще не открыли никакого файла, написано «New Buffer». Чтобы запустить редактор с файлом, просто введите в командной строке имя последнего: **nano filename.txt**. Внизу помещена полезнейшая быстрая справка о наиболее частых сочетаниях клавиш, в формате

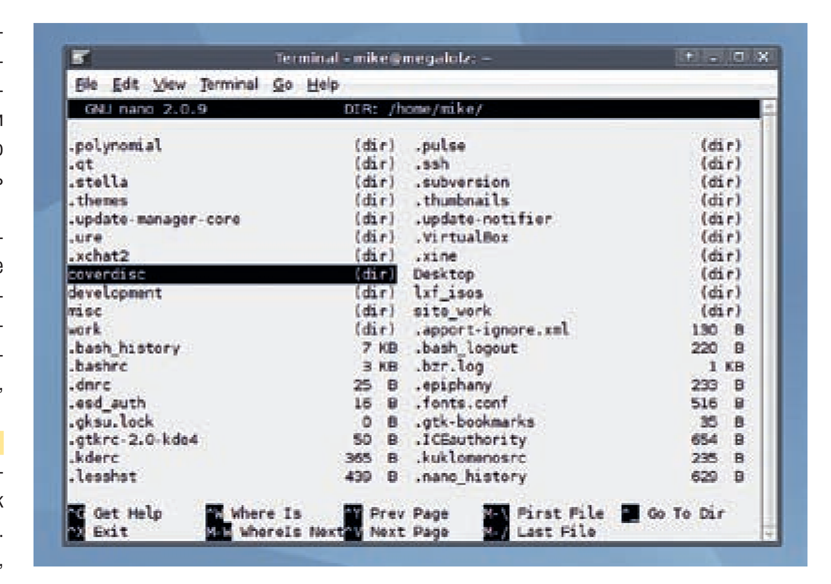
```
^O WriteOut
```

Информация намеренно сжатая, чтобы побольше втиснуть на экран, так что за подробностями обращайтесь к «Quick keybinding reference [Быстрой справке по горячим клавишам]». Здесь символ ^ означает левую клавишу Ctrl вашей клавиатуры, то есть ^O соответствует Ctrl+O (хоть в верхнем, хоть в нижнем регистре). WriteOut сохраняет файл на диск, и нажав Ctrl+O, вы увидите внизу следующее приглашение:

```
File Name to Write:
```

Если вы открывали существующий текстовый файл, то его имя подставится автоматически, и нужно будет только нажать на Enter. Если нет, придется ввести имя самим. Тут мы и убедимся в мощности *Nano*: нажмите Ctrl+t, и экран сменится на прекрасный двухпанельный файловый менеджер. С помощью курсорных клавиш можете перемещаться по каталогам, чтобы найти нужный файл, куда вы сохраните текст.

Как и следовало ожидать, этот файловый браузер пригоден и для других задач. Нажмите Ctrl+X, чтобы выйти из браузера, затем Ctrl+C, чтобы отменить операцию сохранения. Теперь жмите Ctrl+R, чтобы считать файл и вставить результат в текущую сессию редактирования. Когда появится приглашение на ввод имени файла, нажмите Ctrl+t и найдите файл, который требуется вставить.



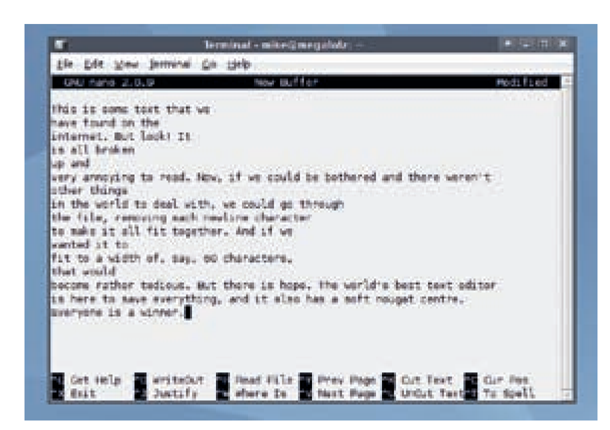
› В *Nano* есть встроенный файловый навигатор, где можно перемещаться курсорными клавишами.

Итак, с открытием, сохранением и вставкой файлов разобрались. Часто используемые сочетания клавиш всегда видны в низу экрана, и вы не запутаетесь. Теперь рассмотрим операции вырезать/вставить. *Nano* в этом отношении сильно отличается от многих других редакторов; однако, попривыкнув, вы поймете, что его способ весьма эффективен.

Введите пять строк какого попало текста, затем переместите курсор на вторую строку. Нажмите Ctrl+K, и эта строка исчезнет. Где именно в строке располагался курсор, неважно: она вся будет вырезана («Kut»). Теперь содержимое этой строки находится в буфере, и можно перейти в любое другое место, нажать Ctrl+U, и строка будет «Uncut» (вставлена в документ).

То же можно проделать и с несколькими строками. Поместите курсор на верхнюю строку нужного фрагмента и нажмите Ctrl+K ровно столько раз, сколько строк нужно вырезать. Прежнее содержимое буфера заменится на новый вырезанный фрагмент. После этого перейдите на то место, куда хотите вставить текст, нажмите Ctrl+U – и все строки, пропавшие при вырезании, появятся в документе. Для нескольких вставок нажмите Ctrl+U нужное число раз.

По умолчанию номера строк не отображаются, однако при нажатии Ctrl+C появится сообщение о статусе, указывающее строку, столбец и символ у текущей позиции курсора, а также смещение от начала файла в процентах от его объема.



› Беда! Взгляните на этот клочковатый текст. Эх, нам бы команду, чтобы он сам переделался в ровный…

Сочетания клавиш

» **Ctrl+X** Выход из редактора. Если вы не сохранили редактированный текст, вас спросят, действительно ли вы хотите выйти.

» **Ctrl+O** Запись (output) текущего содержимого текстового буфера в файл. Появится приглашение ввести имя файла; нажатие Ctrl+T выдаст файловый навигатор, показанный выше.

» **Ctrl+R** Чтение текстового файла и вставка его содержимого в текущую сессию редактора. При запросе на имя файла можно вызвать навигатор, нажав Ctrl+T.

» **Ctrl+K** Вырезание строки в буфер обмена. Несколько нажатий вырежут несколько строк, которые сохранятся как один фрагмент.

» **Ctrl+J** Выравнивание (заполнение) абзаца. По умолчанию при этом текст будет соответствовать ширине окна редактирования.

» **Ctrl+U** Вставка текущего содержимого буфера. Примененная после операции выравнивания, эта команда отменит операцию.

» **Ctrl+T** Проверка орфографии.

» **Ctrl+W** Поиск слова или фразы. В приглашении с помощью курсорных клавиш можете просмотреть предыдущие слова поиска, а нажатие Ctrl+R включит режим замены. Также можно нажать Ctrl+T, чтобы перейти на определенную строку.

» **Ctrl+C** Покажет текущий номер строки и информацию о файле.

» **Ctrl+G** Получить справку; вам выдадут информацию о работе с файлами и основные клавиатурные команды.

Отрада опытных пользователей

N*ano* унаследовал одну из хваленых функций *Pico* – выравнивание текста. При этом текстовый блок встраивается в определенную ширину страницы, посредством переносов и перераспределения слов. Такую возможность включают немногие из текстовых редакторов, тем более легковесных; однако поскольку *Pico* использовался для почтовых сообщений и для чтения групп новостей, где цитаты нескольких уровней требовалось вмещать в экран шириной 80 символов, этой функции уделялось особое внимание.

В новой сессии *Nano* введите следующий текст:

```
сидит кошка
на моем
окошке
```

Теперь представьте, что нужно преобразовать этот текст в одну строку. Во многих текстовых редакторах пришлось бы делать эту работу вручную, переходя на конец строки и нажимая Delete.

А в *Nano* все иначе. Просто поместите курсор на какую-нибудь из этих трех строк и нажмите Ctrl+J, и вы увидите:

```
сидит кошка на моем окошке
```

Внизу экрана возникнет статусное сообщение «Can now UnJustify [Выравнивание можно откатить]». Нажатие Ctrl+U вернет все назад.

По умолчанию *Nano* выравнивает текст по ширине окна редактора. Если вы работаете в *X*-терминале, наберите какой-нибудь текст и с помощью операций вырезать/вставить заполните им половину окна. Нажмите Ctrl+J для выравнивания текста, после чего поменяйте размер окна и нажмите Ctrl+J еще раз. Вы увидите, что текст влился в размер окна. *Nano* выравнивает по абзацам – кускам текста, разделенным пустыми строками.

> «Nano унаследовал одну из хваленых функций Pico.»

Уже неплохо; но настоящая сила прибывает с опцией **-r**. Выйдите из *Nano* и запустите его еще раз с помощью **-r30**. Так вы скажете *Nano*, чтобы все операции выравнивания укладывали текст в ширину 30 символов, независимо от размера окна.

У этой функции есть множество прекрасных применений. Например, британский сайт *Linux Format* ежемесячно рассылает подписчикам письмо с новостями, новыми возможностями и прочим (www.linuxformat.com/newsletters). При этом желательно придерживаться определенной ширины строки письма, чтобы сообщение смотрелось на любой машине. Разные почтовые клиенты по-разному обращаются с простым текстом, и чтобы сделать все как можно проще и надежней, мы запускаем **nano -r68 newsletter.txt**, вводим текст и нажимаем Ctrl+J, форматируя текст для рассылки.

```
Terminal - mike@megalolz: ~
File  Edit  View  Terminal  Go  Help
  GNU nano 2.0.9          File: /usr/share/nano/c.nanorc

## Here is an example for C/C++.
##
syntax "c" "\.(c(c|pp|xx)?|C)$" "\.(h(h|pp|xx)?|H)$" "\.ii?$"
color brightred "\<[A-Z_][0-9A-Z_]+\>"
color green "\<(float|double|bool|char|int|short|long|sizeof|enum|void|static|c$
color green "\<((s?size)|((u_?)?int(8|16|32|64|ptr)))_t\>"
color green "\<(class|namespace|template|public|protected|private|typename|this$
color red "\<(for|if|while|do|else|case|default|switch)\>"
color red "\<(try|throw|catch|operator|new|delete)\>"
color magenta "\<(goto|continue|break|return)\>"
color brightcyan "^[[:space:]]*#[[:space:]]*(define|include|(un|ifn?)def|endif|$
color brightmagenta "'([^'\]|(\\["'abfnrtv\\]))'" "'\\(([0-3]?[0-7]{1,2}))'" "'$
##
## GCC builtins
color cyan "__attribute__[[:space:]]*\(\([^)]*\)\)" "__(aligned|asm|builtin|hid$
##
## String highlighting.  You will in general want your comments and
## strings to come last, because syntax highlighting rules will be
## applied in the order they are read in.
color red "<[^=         ]*>" ""(\\.|[^"])*""
##
                        [ Read 30 lines ]
```

› **Если вы программист, не пожалейте чуток времени на настройку подсветки синтаксиса.**



› **Командой Ctrl+J текст приводится к должному виду. Мир на земле, и в человецех благоволение.**

Другой пример: представьте, что вы нарыли в Интернете замечательную команду с диким количеством параметров – например, команду **mencode** для преобразования видеофайла. Из-за представления web-страницы, при копировании и вставке команда превратится в 20 строк. Вместо того, чтобы удалять все символы перевода строки, просто запустите *Nano* с помощью **nano -r1000**. Вставьте в него текст (почти на всех *X*-терминалах работает Shift+Insert), нажмите Ctrl+J, и весь текст сложится в одну прекрасную строку, пригодную для вставки в терминал. Alt+J выравнивает все абзацы в файле.

Есть и еще возможности. *Nano* умеет автоматически создавать резервные копии файлов после их сохранения. Передайте ему опцию **-B** (**nano -B config_file.txt**), и при нажатии Ctrl+O, помимо сохранения, создастся резервная копия более старой версии файла, с тильдой в конце его имени (**config_file.txt~**). Это стоит проделывать при редактировании важного системного файла – если что-то пойдет не так, старую версию можно восстановить одним взмахом руки.

Полностью поддерживается поиск и замена: нажмите Ctrl+W (Where is…?), и вы увидите приглашение. Введите слово для поиска; а для замены нажмите Ctrl+R. Если в приглашении нажать Alt+R, оно превратится в **[Regexp]**, что позволит выполнять поиск на основе регулярных выражений. С помощью курсорных клавиш вверх и вниз можно перемещаться по более старым строкам поиска – даже тем, что были в предыдущих рабочих сеансах. Они хранятся в вашем домашнем каталоге в файле **.nano_history**.

Интерфейс у *Nano* настраиваемый: по мере привыкания к редактору вам уже не понадобятся быстрые справки внизу насчет сочетаний клавиш. Командой **-x** можно указать *Nano* больше не показывать их. Опция **-c** делает строку состояния видимой всегда, а опция **-S** (S заглавная) заставит *Nano* прокручивать за один раз одну строку, а не страницу. Все это можно объединить:

```
nano -xcSr68 filename.txt
```

Здесь мы задали отсутствие справки, постоянно видимую строку состояния, гладкую прокрутку и выравнивание текста по 68 символам. Прямо как мамочка делывала!

Радости разработчиков

В *Nano* даже есть подсветка синтаксиса для различных языков программирования, хотя по умолчанию она отключена по соображениям производительности: на старых машинах анализ большого кодового файла с тем, чтобы каждая его строка отображалась правильным цветом, требовал много ресурсов. Однако на компьютерах, собранных не более десяти лет назад, вы замедления не заметите.

Чтобы включить подсветку синтаксиса, откройте конфигурационный файл *Nano* (см. врезку ниже) и прокрутите его до последнего раздела. Вы увидите набор строк вида

```
## C/C++
# include "/usr/share/nano/c.nanorc"
```

Знак решетки – это комментарий; удалив его во второй строке, вы предпишете *Nano* считывать на старте файл **c.nanorc** – в нем описана подсветка синтаксиса для файлов исходного кода C и C++. Посмотрев далее по списку, вы увидите, что имеется подсветка для Python, Perl, Ruby, Java, HTML, TeX и для других форматов.

Давайте создадим файл описания подсветки для языка, который не поддерживается в *Nano* по умолчанию. Зайдем в **/usr/share/nano** и скопируем **sh.nanorc** в файл языка, поддержку для которого мы пишем – скажем, в **php.nanorc**. Файл **sh.nanorc** мы выбрали за простоту: его легко модифицировать. А вот, например, в **c.nanorc** вы увидите жуткие нагромождения регулярных выражений, способные просто взорвать мозг.

В файле **php.nanorc** первая незакомментированная строка будет такой:

```
syntax "sh" "\.sh$"
```

Она велит *Nano* использовать данное форматирование для всех файлов, которые заканчиваются на **.sh**, и здесь нужно поменять все вхождения **sh** на **php.** Спуститесь на пару строк ниже, и вы увидите

```
color green "\<(case|do|done|elif|...
```

Когда в исходном коде встречаются ключевые слова **case, do, done, elif** и прочие, они окрашиваются зеленым (**green**). Вы можете удалять ключевые слова и добавлять новые, однако следите, чтобы структура строки не менялась и чтобы ключевые слова отделялись значком канала |. Следующая строка может показаться тарабарщиной, если вы не знакомы с регулярными выражениями:

```
color green "(\{|\}|\(|\)|\;)...
```

Поясним: так задается цвет, используемый для определенных не буквенно-цифровых символов, таких как скобки и точки с запятой. Как и ключевые слова, они отделяются символом канала, но поскольку они и сами являются спецсимволами, применимыми в инструкциях по подсветке синтаксиса, мы «экранируем» их обратной косой чертой.

Если бы мы определили скобки в коде PHP как **{|}|(|)**, это бы не сработало: синтаксический анализатор *Nano* запутался бы, сочтя их инструкциями. Использование обратной косой черты перед каждой из скобок говорит *Nano*, что они должны рассматриваться как отдельные символы. Таким образом, сюда можно подставлять любой не буквенно-цифровой символ, но предварять его знаком \.

```
Terminal - mike@megalolz: ~/old_mike/development/tatanga-0.5.7
File  Edit  View  Terminal  Go  Help
GNU nano 2.0.9            File: tatanga.c

        /* Make proper port number from string grabbed from config file,
         * and do all the other bits and bobs for connection */

        tmp_port = strtol(my_port, (char **) NULL, 10);

        con.sin_family = AF_INET;
        con.sin_port = htons(tmp_port);
        con.sin_addr = *((struct in_addr *) my_host->h_addr);
        memset(&(con.sin_zero), 0, 8);

        if (connect(sck, (struct sockaddr *) &con, sizeof(struct sockaddr)) < 0$
                alert_box("Connect error", "Could not connect to server. Please$
                          "chosen the right server in ~/.tatanga.conf and that th$
                          "network is accessible. Start tatanga with the -n switc$
                          "if you want to reconfigure your settings.");
                exit_app(1);
        }
```

› **Подсветка синтаксиса делает исходный код более понятным.**

Большая часть оставшихся строк в нашем файле **php.nanorc** имеют тот же формат. Ближе к концу вы увидите такую строку:

```
color cyan "(^|[[:space:]])#.*$"
```

Она приказывает анализатору окрасить в голубой (**cyan**) все фрагменты текста, начинающиеся с символа решетки, до конца строки – другими словами, комментарии. В PHP для комментария также применяется двойная косая черта, так что скопируем эту строку и поменяем решетку на \/\/:

```
color cyan "(^|[[:space:]])\/\/.*$"
```

С такой добавкой при открытии файлов PHP вы увидите, что комментарии с двойной чертой тоже раскрашены правильно.

Есть и другие функции, полезные для программистов. При редактировании текста нажмите Alt+I, чтобы включить режим автоматической расстановки отступов: тогда курсор будет оставаться в том же столбце при нажатии на Enter. Это особенно полезно в языках, где пробелы несут синтаксический смысл – например, в Python.

При сохранении файла вам, вероятно, может понадобиться его преобразование в другой формат для другой платформы. Linux и Windows используют разные невидимые символы конца строки, и это порой доставляет много головной боли. Попробуйте, например, открыть текстовый файл, созданный в Unix, с помощью *Блокнота* из Windows XP – все строки слипнутся.

Итак, при появлении приглашения ввести имя файла, нажмите Alt+D, чтобы переключиться между текстовыми стилями Unix и Windows/DOS (Alt+m переведет вас в формат для Mac, однако на самом деле это годится для Mac OS 9 и ранее). Мощь *Nano* позволит брать чьи-то испорченные записи, прокачивать их в текст без переносов строк, а затем выдавать обратно в формате, понятном этим глупым не-Linux-машинам. Ура! LXF

Редактирование файлов настройки

Главный файл настройки *Nano* – **/etc/nanorc**. Но он не одинок: **/etc/nanorc** может содержать другие конфигурационные файлы, например, определения расцветки синтаксиса в **/usr/share/nano**. В **nanorc** можно задать опции по умолчанию, чтобы не прописывать их явно в командной строке. Каждая опция имеет формат:

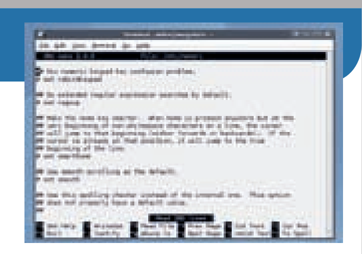
```
# set backup
```

Это эквивалент опции **-B** командной строки. Данная строка закомментирована знаком решетки, и *Nano* будет действовать с настройками по умолчанию, то есть не создавая резервных копий.

Если знак решетки убрать, то по умолчанию создание резервных копий будет включено, а если вы хотите явно сказать *Nano*, чтобы он этого не делал, поменяйте строку вот на какую:

```
unset backup
```

Все опции сопровождаются небольшим текстом-пояснением, так что там не будет ничего непостижимого.



› **Отредактируйте конфигурационный файл, чтобы полнее освоить *Nano*.**

Ядро на Java

Думаете, такой высокоуровневый язык, как Java, не подходит для системного программирования? **Роман Ярыженко** уверит вас в обратном!

LINUX FORMAT на DVD
» Исходные тексты JNode
» NASM 2.08



Наш эксперт

Роман Ярыженко
Всегда готов потратить время и силы на изучение новой операционной системы – особенно если она распространяется вместе с исходными текстами.

Если вы хотя бы изредка следите за событиями в мире ИТ, то наверняка слышали о *Singularity* – исследовательском проекте, являющим собой попытку Microsoft создать полноценную операционную систему на платформе .NET (а если нет, см. http://bit.ly/88zSO). Сама концепция ОС на управляемом коде породила в сообществе множество жарких споров. Мы не будем занимать в них ни ту, ни другую сторону – а просто расскажем вам еще об одном аналогичном проекте. На сей раз он будет свободным (лицензия LGPL) и ориентированным на извечного конкурента .NET (или правильнее будет сказать наоборот?) – платформу Java. Имя ему – JNode, что, в свою очередь, является сокращением от английского «Java New Operating System Design Effort» – «Новая попытка спроектировать операционную систему на Java». Но почему новая?

Сначала было...

Когда появились первые кракеры со своими инструментами, эксплуатирующими уязвимости вроде переполнения буфера, ИТ-общественности пришлось начать думать о том, как защититься от них. К сожалению, язык С, который и по сей день является самым распространенным средством системного программирования, имеет с точки зрения безопасности фундаментальные (конструктивные) недостатки, связанные с использованием указателей. В те далекие времена, конечно, думали в первую очередь о производительности, и конструкция вроде **char buf[256]; scanf("%s», buf);** мало кого смущала. Впрочем, и когда взломщики стали неизбежным злом, разработчики средств защиты не отказались от С совсем, а лишь попытались сделать его надежнее. Это, безусловно, смягчило проблему, но не решило ее до конца.

В середине 90-х годов теперь уже прошлого столетия теперь уже бывшая компания Sun Microsystems представила платформу Java с одноименным языком программирования. Последний позиционировался Sun как кросс-платформенный, легкий (думаю, вы можете с этим поспорить) и безопасный. Справедливости ради надо отметить, что Sun не была здесь пионером – языки со схожими чертами предлагались и до нее. Но они, по причине ли плохого маркетинга, либо по какой-то другой, не получили широкого распространения. С самого момента создания Java предпринимались попытки перенести на нее все необходимые приложения, чтобы создать операционную среду. Сначала это была JavaOS, потом – JBS, и, наконец, появился JNode.

Цель разработчиков JNode, заявленная на сайте проекта (www.jnode.org) – это «создание на базе Java простой в применении и установке операционной системы для личного использования. Любое Java-приложение должно запускаться быстро и безопасно».

Первая публичная версия JNode была выпущена в середине 2003 года. На момент написания статьи, последний официальный релиз имел номер 0.2.8 и датировался январем 2009 года. За прошедшие полтора года в системе, несомненно, произошли некоторые изменения, поэтому мы будем ориентироваться на «ночные сборки», на самом деле представляющие собой ежесуточные снимки Subversion-репозитория JNode.

Действительно просто

Поскольку история проекта (пока еще!) не очень длинна и не изобилует интересными подробностями, останавливаться на ее деталях мы не будем, и перейдем непосредственно к сборке JNode. Можно, конечно, взять готовый образ, но гораздо интереснее скомпилировать все самостоятельно из исходных текстов. Для этих целей нам понадобятся:

» Собственно исходные коды (их можно загрузить с сайта http://www.jnode.eu/releases/nightly-builds/jnode_svn_sources.tbz или взять **LXF**DVD).

» JVM и JDK 1.6 (подойдет и OpenJDK, и официальная сборка от Sun – как минимум одна из них должна присутствовать в репозиториях вашего дистрибутива).

» Ассемблер nasm (www.nasm.us).

«Но постойте! Разве *GCC* не нужен?» – вероятно, спросит грамотный читатель. Как это ни странно, нет. JNode почти полностью написан на Java – исключение составляют порядка тридцати файлов на ассемблере. Как же у разработчиков получается увязать между собой низкоуровневый язык и Java? Все дело в том, что после компиляции с помощью *javac* часть байт-кода, которая, собственно, и есть ядро (если его можно так назвать), преобразуется в родной код для процессоров x86 или x64. Все это, конечно, сродни «черной магии», в основе которой лежит специальный JIT-компилятор байт-кода, созданный разработчиками JNode – L1A (Level 1A). Чтобы отличить, вызывается ли он во время сборки или же во время выполнения, используется специальная переменная. Поскольку код, составляющий загрузочный образ, не задействует динамическую составляющую языка Java, компи-

```
mem-size 2FEF0000 init page-dir table
enable paging
paging setup finished
Version 0.2.9-dev December 28 2009
Before start_vm
VmArchitecture32#boot
Memory map
Available    00000000 00000000 - 00000000 0009F7FF
Reserved     00000000 0009F800 - 00000000 0009FFFF
Reserved     00000000 000DC000 - 00000000 000FFFFF
Available    00000000 00100000 - 00000000 2FEEFFFF
ACPI reclaim 00000000 2FEF0000 - 00000000 2FEFEFFF
ACPI NVS     00000000 2FEFF000 - 00000000 2FEFFFFF
Available    00000000 2FF00000 - 00000000 2FFFFFFF
Reserved     00000000 FEC00000 - 00000000 FEC0FFFF
Reserved     00000000 FEE00000 - 00000000 FEE0FFFF
Reserved     00000000 FFFE0000 - 00000000 FFFFFFFF
Initialize MemoryBlockManager
Start end: 07C30000 2FEF0000
Size     : 282C0000
end of initialize.
[loadFromBootClassArray:]
Starting JNode
VmSystem.initialize
```

› **Понимая внутреннее устройство JNode, вы сможете расшифровать эти сообщения, если сумеете прочесть их с экрана.**

ляция выполняется не сложнее, чем в случае того же *GCC* (ведь фундаментальные отличия между C и Java кроются не в синтаксисе, а в управляемой природе последней). Правда, с оптимизацией у L1A пока еще не очень хорошо, но для проекта версии 0.2 это, в общем, простительно.

Предположим, вы скачали и распаковали исходный код, а также установили необходимые пакеты. Что теперь? Находясь в каталоге с исходными кодами, скомандуйте:

```
./build.sh
```

Пока идет сборка, давайте рассмотрим особенности ОС.

JNode изнутри

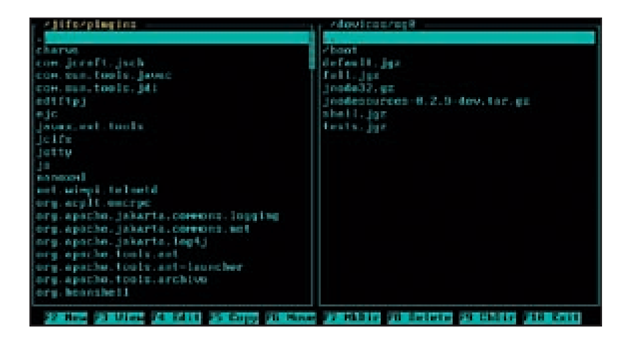
Это может показаться странным, но у JNode нет ядра в привычном понимании этого слова. Операционная система имеет трехуровневую структуру. В самом низу расположены виртуальная машина Java (ВМ) и ее базовые компоненты: например, сборщик мусора. Часть ВМ, написанная на ассемблере, по каким-то причинам называется «наноядром». Это не вполне точно, так как «наноядро» не может работать в отрыве от ВМ (в частности, ассемблерный код вызывает методы, которые на нем не реализованы), но, в конце концов, это вопрос терминологии. Выше расположены библиотека классов, драйверы, файловые системы и (по желанию) GUI. Как можно писать на Java драйверы, требующие низкоуровневого взаимодействия с оборудованием – например, реакции на прерывания? Все просто. Драйвер регистрирует в ассемблерном коде обработчик IRQHandler, а при возникновении прерывания увеличивается их счетчик, после чего планировщик распределяет каждое прерывание на соответствующие потоки с обработчиком. Еще выше находятся пользовательские приложения.

Это может показаться странным, но JNode не пользуется аппаратными кольцами x86, выполняя весь код в привилегированном режиме ring 0. Безопасность обеспечивается средствами платформы Java, производящей проверку байт-кода как при загрузке, так и при выполнении. Следовательно, прямой доступ к памяти пользовательских приложений и ВМ получить затруднительно. Также в JNode нет понятия «процесс», поскольку в Java они не поддерживаются. Но зато есть потоки и так называемые «Isolates»; последние можно считать аналогом процессов. Из дисковых файловых систем в JNode в данный момент поддерживаются (в разной степени) FAT, ext2, HFS+, NTFS, ISO9660, из сетевых – NFS и CIFS. Есть еще несколько псевдофайловых систем, таких как, например, JIFS (аналогичен proc в Linux).

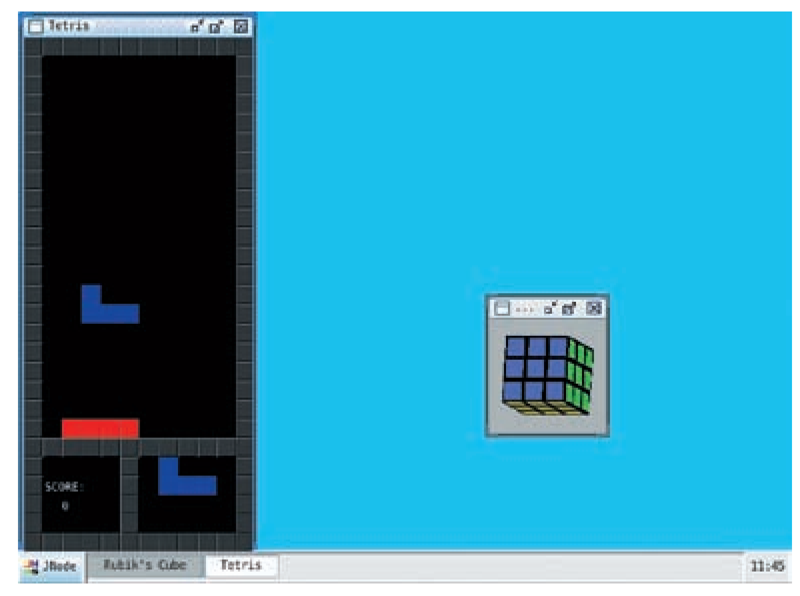
> «Безопасность обеспечивается средствами платформы Java.»

Для пользователя

Но что это? JNode уже скомпилировался! ISO-образ системы можно найти в каталоге **путь_к_JNode/all/build/cdroms/**, но желая попробовать его в виртуальной машине, имейте в виду, что в *VirtualBox* он работает очень медленно, а в *Qemu* не поддерживается графика. Увы, разработчики ориентируются на *VMware*.



› Файловый менеджер CC понравится пользователям MC.



› В JNode нет ни офисного пакета, ни базы данных, зато есть ИГРЫ! Это практически идеальная система для офиса.

В качестве загрузчика в JNode используется *Grub*. Рекомендую выбрать пункт «JNode (all plugins)», с ним можно запустить больше приложений. Старт системы можно видеть на рисунке: все, что появляется после строки «before start_vm», выводится откомпилированным кодом Java.

Увидев стандартное приглашение, можно нажать Tab для получения списка доступных команд. Для некоторых из них доступно даже автодополнение ключей. Команды достаточно знакомые (*grep*, *find*, *tar* и пр.), но поклонников Norton Commander ждет приятный сюрприз – файловый менеджер *cc* (нет, это не компилятор C).

Графический интерфейс, хоть и в зачаточном состоянии, в JNode тоже имеется. Подсистема рассчитана на *VMware*; имеются драйверы для старых видеокарт ATI и Nvidia. Прежде чем активировать GUI, наберите **gc** (принудительная сборка мусора, может занять секунд пять), а после этого – startawt. Потребуется некоторое время, чтобы графическая подсистема запустилась; возможно, надо будет пошевелить мышью. Интерфейс... в нем нет ничего особенного, разве что он аскетичен и иногда «притормаживает». Зато в меню есть пара простеньких игр.

Что же дальше?

На данный момент (после семи лет разработки), JNode находится примерно на уровне ранних версий ядра Linux. Нет многих драйверов устройств, не реализованы утилиты командной строки, графическая и сетевая подсистема находятся в зачаточном состоянии... Все это говорит, во-первых, о малочисленности разработчиков, а во-вторых... JNode, в общем-то, нельзя назвать ОС – это скорее концепция.

Вычислительные мощности современных компьютеров позволяют выполнять управляемый код практически без проблем с производительностью. Разработчики JNode пошли еще дальше и решили, что на Java можно писать не только приложения, но и саму ОС. Это значит, что такая система, если она когда-нибудь достигнет промышленного уровня, будет надежнее и безопаснее, нежели ее «неуправляемые» аналоги. LXF

Go: новый уровень

В прошлый раз мы разобрались в основах синтаксиса Go. Сегодня **Андрей Боровский** займется более сложными аспектами языка.



Наш эксперт

Андрей Боровский
Постоянно открывает для себя что-то новое, изучая языки, библиотеки и прочие инструменты программиста по заданию **LXF**.

Является ли богатство синтаксических конструкций признаком хорошего языка программирования? Я думаю, нет. Очень часто это многообразие возникает в результате того, что изначальный синтаксис был не очень удачным. Стремясь исправить положение, разработчики добавляют новые формы, но старые, уже используемые, удалить не могут. Впрочем, избыточность свидетельствует и о долгой истории развития языка. Go молод, и его разработчики имеют богатый практический опыт, а значит, утверждая, что в Go отсутствуют «пережитки» C и C++, они, скорее всего, правы. Возможно, именно поэтому скромный набор базовых средств языка Go не создает при работе ощущения скованности. С другой стороны, лаконичный синтаксис Go лишает нас удовольствия поломать голову над конструкциями типа

```
i[++i] = ++i [i++]
```

(операторы инкремента и декремента присутствуют в Go только в постфиксной форме).

Ваш пакет

Строго говоря, мы уже создавали свой пакет месяц назад – любая Go-программа состоит минимум из одного. Синтаксис описания разделяемых пакетов тот же самый; нужно лишь помнить, что по правилам Go экспортируемые из пакета идентификаторы должны начинаться с заглавной буквы. Для примера рассмотрим пакет *threads*, содержащий набор функций для работы с потоками.

```
package threads
var c = make(chan int, 1)
var counter int = 0
func NewThread() {
        counter++
}
func ExitThread() {
        counter--
        c <- 1
}
func WaitForThreadsLessThan(n int) {
        for counter >= n {
                <- c
        }
}
func Join() {
        WaitForThreadsLessThan(1)
}
func ThreadsCount() int {
        return counter
}
```

Функция WaitForThreadsLessThan() приостанавливает выполнение вызвавшего ее потока до тех пор, пока число запущенных потоков (переменная **counter**) не станет меньше заданного значения. Для этого используется механизм синхронизации с помощью буферизованных каналов (**LXF130**). Чтобы это работало, перед запуском каждого нового потока нужно вызвать NewThread(), а перед завершением – ExitThread().

Протестируем наш пакет на простой многопоточной программе – сетевом сервере (да, в Go многопоточный сетевой сервер действительно может быть простой программой).

```
package main
import (
    net "net"
    flag "flag"
    fmt "fmt"
    threads "./threads"
 )
const max_threads int = 2
func reply(connection * net.TCPConn) {
    buffer := make([]byte, 255)
    connection.Read(buffer)
    connection.Write(buffer)
    connection.Close()
    threads.ExitThread()
 }
func main() {
    flag.Parse()
    if flag.NArg() != 1 {
            fmt.Printf("Использование: ./server <address>\n ");
    }

    addr, e := net.ResolveTCPAddr(flag.Arg(0) + ":7")
    if e != nil {
            fmt.Printf("Ошибка привязки адреса (%s)", e)
            return
    }
    sock, e := net.ListenTCP("tcp", addr)
    if e != nil {
            fmt.Printf("Ошибка инициализации сервера (%s)", e)
            return
    }
    for {
            threads.WaitForThreadsLessThan(max_threads)
            conn, e := sock.AcceptTCP()
            if e != nil {
                    fmt.Printf("Ошибка соединения (%s)", e)
                    return
            }
            threads.NewThread()
            go reply(conn)
    }
}
```

Наш сервер слушает порт 7 (стандартный для протокола ECHO). Получив от клиента запрос на соединение, сервер считывает его первые 255 байт, отправляет их обратно клиенту и закры-

вает соединение (это не совсем соответствует стандарту протокола ECHO, но нам хватит). Адрес для привязки сервера передается программе в виде аргумента командной строки.

Обмен данными с клиентом выполняет функция reply(), которую мы вызываем как сопроцедуру. Пакет *threads* нужен для того, чтобы ограничить число одновременно выполняющихся потоков. Использование в программе нашего пакета ничем не отличается от использования стандартного – например, *fmt*. Единственное, что нужно учесть – это расположение файла пакета. Если он находится не в стандартной директории пакетов Go, в строке импорта пакета мы должны указать полный путь к нему, даже если файл пакета находится в той же директории, что и файл программы. Пакет должен быть скомпилирован в объектный код (без вызова компоновщика) перед компиляцией основной программы.

Пакеты могут состоять из нескольких файлов исходных текстов Go (расположенных в одном каталоге), при этом их имена (за исключением одного, естественно) не могут совпадать с именем пакета.

Помимо пакета *threads*, в нашей программе-сервере используются и другие. *Net* содержит базовые функции, необходимые для работы с сетью. *Flag* предоставляет средства для анализа командной строки.

Сложные типы

Помимо примитивных типов данных, в Go имеется и ряд сложных. Наряду с уже известными нам каналами, к ним относятся строки, массивы [arrays] и их сечения [slices], структуры, интерфейсы, ассоциативные массивы [maps] и процедурные типы.

Переменные-массивы в Go объявляются так же, как и переменные простых типов:

```
var a [32] byte
```

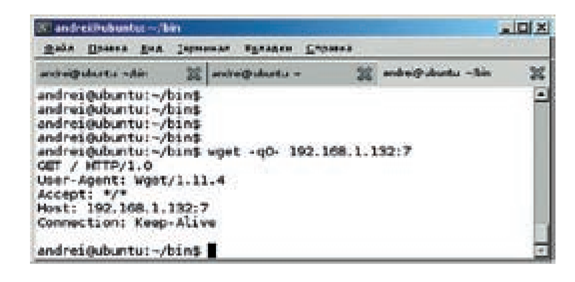
Массивы в Go не являются указателями, как в C, однако память, выделенная массиву элементов простого типа, расположена непрерывно. Таким образом, массив Go можно «конвертировать» в массив C с помощью указателя на первый элемент (это правило важно при работе с функциями, импортируемыми из библиотек C, и оно не распространяется на сечения, которые нельзя преобразовать в массивы C).

Сечения похожи на массивы, за исключением того, что при объявлении переменной этого типа длина массива не указывается. Сечение может быть создано динамически с помощью функции make(). Важное применение сечений – передача массивов в функции. Сечения и массивы совместимы между собой, так что функции, принимающей сечение, можно передать любой массив заданного типа. Эффективность сечений как параметров функций объясняется тем, что это ссылочный тип, который можно рассматривать как указатель на фрагменты массивов с особым синтаксисом. До тех пор, пока значение переменной-сечения не проинициализировано, она равна **nil**.

Структуры Go очень похожи на таковые в C. Они могут объединять переменные и функции.

```
struct {
        x, y int
        u float
        A *[]int
        F func()
}
```

Любопытной особенностью Go является возможность включить в состав структуры «переменные набивки» (в качестве их имени указывается символ _, и обратиться к ним нельзя). Их используют для выравнивания адресов членов структуры по определенным границам, что может понадобиться, например, при взаимодействии с кодом, написанным на C. Скажем, если структура Go содержит двухбайтные переменные, а правила выравнивания в коде C требуют, чтобы адреса всех переменных-членов структуры были кратны 4, поля в структуре Go можно выровнять с помощью переменных набивки.



› Многопоточный ECHO-сервер на Go: действительно, работает.

Те, кому приходилось использовать код C из других языков программирования, знают, сколько проблем могут создать различия в правилах выравнивания адресов переменных в структурах. Однако стоит отметить, что использование переменных набивки не является универсальным решением проблемы. Разные компиляторы на разных платформах могут использовать разные правила выравнивания, так что сопряженный с C код Go может работать на одной платформе и не работать на другой.

Еще одна интересная особенность структур – безымянные переменные. При их объявлении указывается только тип, но не имя. Безымянными членами структур обычно являются другие структуры. Пусть у нас есть некоторая структура t, и мы объявляем структуру S как

```
type S struct {
        t
        ...
}
```

В этом случае все поля и методы t можно вызывать как поля и методы S. Если вы пользовались *GTK+*, эта концепция должна быть вам знакома.

Интерфейсы похожи на структуры, но они объединяют только функции. Их часто используют в пакетах для экспорта. Вот, например, фрагмент файла **net.go** (является частью пакета *net*)

```
type Conn interface {
  Read(b []byte) (n int, err os.Error)
  Write(b []byte) (n int, err os.Error)
  Close() os.Error
  LocalAddr() Addr
  RemoteAddr() Addr
  SetTimeout(nsec int64) os.Error
  SetReadTimeout(nsec int64) os.Error
  SetWriteTimeout(nsec int64) os.Error
}
```

Интерфейсы используются в пакетах «не для красного словца». Они позволяют инкапсулировать объекты, о чем будет сказано ниже.

Интерфейсы представляют собой нечто большее, чем просто наборы функций. Пусть I – некоторый интерфейс. Тогда »

Тонкости многопоточности

Незабвенный доктор Эмет Браун советовал Марти мыслить в четырех измерениях. Работа с потоками также требует особой культуры мышления. На первый взгляд может показаться, что функцию NewThread() следует вызывать первой строчкой в теле функции reply() (такой код смотрелся бы красивее), но в этом случае функция WaitForThreadsLessThan() не смогла бы гарантировать, что число потоков действительно не превышает заданного значения (подумайте, почему).

Многопоточность в Go – настоящая, а не корпоративная, как, например, в Lua, однако количество потоков, которые использует программа Go, как правило, меньше, чем число выполняемых сопроцедур. Разработчики Go попытались найти компромисс между использованием потоков для реализации сопроцедур и накладными расходами на создание нового потока в операционной системе. Предложенное ими решение заключается в том, что несколько сопроцедур разделяют на уровне ОС один поток. Если одна из сопроцедур блокирует поток, остальные связанные с ним сопроцедуры могут быть перенесены в другой. Все это, напомним, происходит в скомпилированной программе по ходу ее работы. Иными словами, программам Go требуется свой менеджер управления потоками времени выполнения. Разработчику обычно не приходятся задумываться о таких вещах, но, поработав с сопроцедурами, вы заметите, что переключение между ними обычно происходит в момент блокировки или при вызове таких функций как fmt.Printf().

переменная типа I будет совместима с переменными других типов при условии, что множество методов интерфейса I является подмножеством методов этих типов.

Ассоциативные массивы Go позволяют выполнять индексирование переменными различных типов, а не только целочисленными. Например,

```
a := map [string] int
```

представляет собой ассоциативный массив целочисленных переменных, индексируемых строками. В качестве индексов ассоциативного массива нельзя применять структуры и массивы, так как для них не определена операция проверки равенства. Интерфейсы же могут выступать в роли индексов, но при этом операция равенства для них должна быть реализована. Ассоциативные массивы отличаются от обычных не только типами индексов: обычному массиву соответствует непрерывная область памяти, а для ассоциативных массивов это, вообще говоря, не выполняется.

Объекты в Go

Go не является объектно-ориентированным языком, однако при необходимости в нем можно создавать нечто вроде объектов. Для примера напишем альтернативную реализацию пакета *threads* (файл **oopthreads.go**). Один из недостатков приведенного выше кода заключается в использовании глобальных переменных. Глобальные переменные – не проблема, если нам нужен только один менеджер сопроцедур; но, предположим, требуется управлять несколькими группами сопроцедур так, чтобы для каждой из них была определена своя функция Join() и т.д. Одно из возможных решений – использование структур с методами:

```
package oopthreads
type Thread struct {
        c chan int
        counter int
}
func NewThreadObject() * Thread {
        t := new(Thread)
        t.c = make(chan int, 1)
        return t
}
func (this * Thread) NewThread() {
        this.counter++
}
func (this * Thread) ExitThread() {
        this.counter--
        this.c <- 1
}
func (this * Thread) WaitForThreadsLessThan(n int) {
        for this.counter >= n {
                <- this.c
        }
}
func (this * Thread) Join() {
        this.WaitForThreadsLessThan(1)
}
func (this * Thread) ThreadsCount() int {
        return this.counter;
}
```

Переменные, необходимые для управления потоками, собраны в структуру *Thread*, но это не главное. Функции, управляющие сопроцедурами, теперь объявлены как методы этой структуры. Объявление функции-метода для некоторого типа, помимо стандартного заголовка, включает в себя параметр-указатель на значение данного типа. Этот параметр объявляется перед именем функции. Например, объявленная выше функция

```
func (this * Thread) ThreadsCount() int
```

есть метод структуры Thread, а

```
func ThreadsCount(this * Thread) int
```

– нет. Методы структуры являются ее селекторами, то есть их следует вызывать, используя синтаксис

```
<имя_переменной_структуры>.<имя_метода>
```

При этом в дополнительном параметре метода передается указатель на переменную, для которой он вызван. Он используется в теле метода для обращения к значению указанной переменной. Описанное очень похоже на реализацию объектов в объектно-ориентированных языках программирования, за исключением того, что в них параметр this создается автоматически.

Особого внимания заслуживает проблема конструкторов. Для динамического создания структур мы должны пользоваться функцией new(), заполняющей выделенный участок памяти нулями. Функцию new() нельзя совместить с вызовом конструктора, как в C++, так что проблему конструкторов придется решать отдельно. Для инициализации структуры можно создать специальный метод, например:

```
func (this * Thread) init() {
        this.c = make(chan int, 1)
}
```

В этом случае экземпляр структуры нужно сначала создать с помощью функции new(). В нашем примере используется другой распространенный прием: функция-фабрика объектов NewThreadObject(), возвращающая указатель на уже готовый объект.

Поскольку объект в Go – обычная структура, переменные-объекты можно создавать и статически, и даже инициализировать их в процессе объявления – например, переменная типа Thread может быть объявлена и проинициализирована такой строкой:

```
t := Thread {make(chan int, 1), 0}
```

Перепишем наш сервер с использованием пакета *oopthreads*:

```
package main
import (
   net "net"
   flag "flag"
   fmt "fmt"
   . "./oopthreads"
)
const max_threads int = 1;
var thread_manager * Thread
func reply(connection * net.TCPConn) {
   buffer := make([]byte, 255)
   connection.Read(buffer)
   connection.Write(buffer)
   connection.Close()
   thread_manager.ExitThread()
}
func main() {
   flag.Parse()
   if flag.NArg() != 1 {
      fmt.Printf("Использование: ./server <address>\n ");
```

Функции make() и new()

Мы уже встречались с функцией make() в нескольких примерах. Она предназначена для динамического создания каналов, сечений и ассоциативных массивов (следует помнить, что значение, возвращаемое make(), является не указателем на область памяти, а значением переменной). Функция make() не только выделяет область памяти для новой переменной, но и инициализирует ее. В общем виде вызов make() выглядит так:

```
make(<тип_данных>, <значение инициализации> [,<значение инициализации>] ...)
```

Параметры инициализации зависят от типа создаваемой переменной.

Функция new() также используется для динамического создания переменных, но работает иначе, чем make(). Единственный аргумент new() – тип создаваемой переменной. Вызов new(T) возвращает значение типа *T. Выделенная под переменную область памяти заполняется нулями, и никакой дальнейшей инициализации не производится.

```
    }
    thread_manager = NewThreadObject()
    addr, e := net.ResolveTCPAddr(flag.Arg(0) + ":7")
    if e != nil {
        fmt.Printf("Ошибка привязки адреса (%s)", e)
        return
    }
    sock, e := net.ListenTCP("tcp", addr)
    if e != nil {
        fmt.Printf("Ошибка инициализации сервера (%s)", e)
        return
    }
    for {
        thread_manager.WaitForThreadsLessThan(max_threads)
        conn, e := sock.AcceptTCP()
        if e != nil {
            fmt.Printf("Ошибка соединения (%s)", e)
            return
        }
        thread_manager.NewThread()
        go reply(conn)
    }
}
```

Обратите внимание, что в программе, импортирующей модуль *oopthreads*, мы не можем получить доступ к членам структуры Thread.c и Thread.counter, поскольку имена этих переменных начинаются со строчной буквы. Таким образом достигается некоторая инкапсуляция.

Методы могут быть не только у структур. Взгляните, например, на такой код:

```
type Sequence []int
func (s Sequence) Len() int {
  return len(s)
}
func (s Sequence) Less(i, j int) bool {
  return s[i] < s[j]
}
func (s Sequence) Swap(i, j int) {
  s[i], s[j] = s[j], s[i]
}
```

Хотя определение метода требует, чтобы дополнительный параметр был указателем, в приведенном выше примере символ * отсутствует – потому что в определении типа Sequence используется сечение, которое является указателем автоматически.

Необходимо различать методы и функции-члены структур. Методы – это функции, связанные с определенным типом данных, в то время как функции-члены являются обычными переменными типа «указатель на функцию», и им могут быть присвоены любые функции, не связанные ни с каким конкретным типом.

Важную роль в реализации объектов Go играют интерфейсы. Любой тип данных, связанный с некоторым набором методов, можно привести к типу «интерфейс», функции которого реализуют некоторое подмножество множества этих методов. Например, можно модифицировать пакет *oopthreads*, добавив в него

```
type IThread interface {
    NewThread()
    ExitThread()
    WaitForThreadsLessThan(n int)
    Join()
    ThreadsCount() int
}
func NewIThread() * IThread {
    i := IThread(NewThreadObject())
    return &i
}
```

Функция NewIThread() сначала создает объект Thread, а затем приводит его к типу IThread. В результате мы достигаем полной инкапсуляции объекта Thread. Если программы, импортирующие модуль *oopthreads*, используют исключительно интерфейс IThread, мы можем заменить объект Thread на любой другой со схожим набором методов совершенно незаметно для этих программ. Интерфейсы позволяют нам реализовать для объектов Go некое подобие полиморфизма. Функция, предназначенная для работы с объектами, родственными Thread, может быть объявлена как

```
func Foo(thread * IThread)
```

› **Так выглядит вывод программы 6nm для модуля threads.**

Другие языки

Импортировать функции из библиотек C в программу, написанную на Go, не просто, а очень просто. Правда, для этого потребуются специальные инструменты. Помимо набора утилит для работы Go, которым мы пользовались до сих пор, существует еще один инструментарий, основанный на GCC. Программа *gccgo* представляет собой головную часть системы GCC, добавляющую Go к набору языков, поддерживаемых *GNU Compilers Collection*. Для работы с *ggcgo* вам придется выкачать из репозитория и самостоятельно собрать проект из исходных текстов. Потребуется также специальный компоновщик *gold* (аналог *ld* для Go). Вместе с *gccgo* компилируются и устанавливаются некоторые другие утилиты, также необходимые для сборки. Поскольку их имена совпадают с названиями их стандартных аналогов, я рекомендую устанавливать *gccgo* в отдельную директорию. После того, как все будет настроено, написать программу, использующую функции C, становится действительно просто. Вот как, например, импортируется функция **perror()** из стандартной библиотеки C:

```
package main
func c_perror(s * byte) __asm__ ("perror")
func main() {
    var msg = [5]byte{'t', 'e', 's', 't', 0}
    c_perror(&msg[0])
}
```

Имя импортируемой функции указывается в скобках после ключевого слова __asm__ (что похоже на синтаксис встраиваемого ассемблера в *GNU C*, но отличается по сути). Типу char * из C в Go соответствует массив типа byte. При вызове функции мы передаем ей указатель на первый элемент массива (сечения использовать нельзя). Простые типы вроде int и byte, а также структуры и указатели могут свободно передаваться между C и Go; остальные типы несовместимы. Если заглянуть под капот Go, мы увидим, что сечениям и строкам Go соответствуют простые структуры на C. В принципе, можно написать C-функцию, которая, получив указатель, например, на переменную-сечение Go, будет работать с соответствующей структурой C; однако внутреннее представление типов Go может меняться.

Существует и механизм экспорта функций Go в программы, написанные на C, но он еще не обрел законченной формы. С помощью *gccgo* можно преобразовывать объявления типов, переменных и функций C в объявления Go (для этого программу нужно запускать с ключами **-S --ggo**), однако механизм преобразования все еще далек от совершенства.

Теперь вы знаете достаточно, чтобы начать применять Google Go в своей работе. Конечно, язык пока далек от завершения, и, возможно, выйдет из лабораторий не раньше Perl 6, но все же, если вы реализуете на нем что-то стоящее – дайте нам знать. LXF

Что за штука... MeeGo?

Если совместить операционные системы Moblin и Maemo, что получится? MeeGo, конечно! **Пол Хадсон** объяснит, что все это значит.

» Среди дурацких названий ПО для Linux MeeGo, по-моему, вне конкуренции. И кто его только придумал?
Ну вы уж скажете! Да, имя необычное. Согласен, дурацкое. Это новый проект, объединивший Moblin (тоже лишено сладкозвучия) и Maemo («мэймо» это, «маимо» или «мимо» – спорят по сей день). По-моему, MeeGo – вполне достойный венец этому союзу.

» Быть может, это попытка создать броское имя-брэнд в стиле Web 2.0...
На самом деле, нет. Само имя отнюдь не новое. Фонд Linux Foundations зарегистрировал домен meego.com лет десять назад. Не знаю, что было у них на уме, но они владеют этим товарным знаком уже давненько.

» Об имени довольно. Что за ним стоит? Какова подоплека?
MeeGo объединяет грандиозную работу, проделанную Intel для Moblin, с усилиями Nokia по разработке Maemo. Поэтому, прежде чем мы начнем рассуждать о новом проекте, необходимо уяснить, что в основе MeeGo лежит мощный пласт ПО, инструментов, разработок и документации существующих проектов.

> «Не ждите, что Nokia вмиг переведет свои аппараты на MeeGo.»

» Тогда что, например, внес в проект Moblin?
Нравится нам это или нет, но именно компания Intel спровоцировала нетбуковую революцию. Усилия, вложенные в разработку проекта One Laptop Per Child («Ноутбук каждому ребенку»), привели к удешевлению производства компьютерных компонентов. Если собрать из этих компонентов ноутбук – получится нетбук. На волне успеха в Intel разработали процессор Atom повышенной энергоэффективности.

» Значит, Intel вкладывается в основном по аппаратной части?
Да, Intel известный производитель чипов. Но проект Moblin включает немало ПО. Помнится, несколько лет назад Apple высказывался в таком духе: чтобы людям понравилось ваше «железо», им должно понравиться и ваше ПО. Поэтому Intel приложил огромные усилия, чтобы оптимизировать и приспособить Linux к своим изделиям. В результате Moblin быстро загружается, занимает мало места на накопителе и обладает настраиваемым пользовательским интерфейсом, которому и на маленьком экране не тесно. По большей части этот интерфейс написан с помощью *Clutter* – инструментария на основе *GTK*, идеального для создания рисунков и анимации. Если вы когда-либо видели аккуратный интерфейс Ubuntu Netbook Remix, так это был *Clutter*. Все эти ресурсы Intel вкладывает в MeeGo.

» Вот здорово! А что же со стороны Maemo?
По ряду причин Maemo не столь успешен, как Moblin. Во-первых, долгое время система устанавливалась только на интернет-планшеты и коммуникаторы Nokia высшего ценового диапазона – N770, N800, N810 и, совсем недавно, N900. Машины превосходные, но для узкой ниши; и на рынке их, по сравнению с нетбуками, очень мало. Кроме того, до недавнего времени Maemo базировался на *GTK* с собственными модификациями Nokia. Но пару лет назад, когда Nokia приобрела Trolltech, фирму-создателя *Qt*, сделалась очевидной неизбежность перехода Maemo на *Qt*.

» И что, Maemo перешел на *Qt*?
Не совсем... Хотя приобретение состоялось в 2008-м, поддержка *Qt* появилась в Maemo только в прошлом году. А ведь поддержка отнюдь не означает использования в самой системе – здесь все по-прежнему кодируется на *GTK*. В результате Maemo содержит и *GTK*, и *Qt*, что не есть изящный подход для маленьких устройств.

» Так если у Maemo есть *GTK* и *Qt*, а у Moblin только *GTK* – почему бы им не сойтись на *GTK*?
Правильный вопрос, но ответа у меня, к сожалению, нет...

» Ну, тогда продолжайте...
Увы, вместо совместного использования общих достижений, за основу интерфейса MeeGo решено взять *Qt*.

» Значит, *Clutter* приказал долго жить?
Как я уже упоминал, Intel вкладывает в MeeGo все свои ресурсы, включая *GTK* и *Clutter*. Однако проект MeeGo ясно дал понять, что они включены только для совместимости – чтобы обеспечить работу существующих приложений – и что основным инструментарием будет *Qt*. Учитывая, что *Qt* не стал базой даже для Maemo, такой ход трудно назвать логичным. Но поскольку Nokia владеет *Qt*, она отчасти обязана его использовать.

» Правда, здорово, когда такие киты, как Nokia, переходят на Linux?
Всегда приятно, когда нашего полку прибывает – но не ждите, что Nokia враз переведет все свои аппараты на MeeGo. У фирмы солидная доля рынка в низшем ценовом сегменте – а ведь там телефоны недорогие, с маленькими экранами и слабыми процессорами. Вполне естественно ставить полноценные Linux-дистрибутивы на мощные современные аппараты – но не на телефоны, прилагаемые в подарок к договору на обслуживание.

» У меня есть такой, и он мне нравится – в конце концов, это всего лишь телефон, а не браузер, не почтовый клиент и не MP3-плейер...
Верно, но в таких ситуациях разумнее применить Symbian, современный потомок операционной системы EPOC от Palm, которая ныне тоже принадлежит Nokia. На этой системе с открытым исходным кодом работает едва ли не половина современных смартфонов. Кроме того, *Qt* уже присутствует на Symbian-устройствах – а значит,

до базовой совместимости приложений MeeGo и Symbian рукой подать.

Значит ли это, что со временем, когда телефоны станут мощнее, проект MeeGo распространится на модели низшего ценового диапазона Nokia?
Весьма вероятно. Определенно в интересах Nokia воспользоваться потенциалом оптимизации, полученным от проекта Moblin, и продвигать MeeGo на все более мелкие аппараты.

Если приложения будут работать на любых устройствах от класса Symbian до нетбуков, как программисты смогут определять назначение своих программ?
Больная тема. Например, все модели iPhone оснащены экранами 320×480, и все приложения работают и выглядят одинаково. В случае с MeeGo заранее не известно ни разрешение экрана, ни объем ОЗУ, ни даже тактовая частота процессора.

Интересное замечание, но разве разница между процессорами 1,6 ГГц и 1,8 ГГц так уж велика?
Вы правы, но MeeGo предназначается еще и для разных архитектур. То есть ПО сможет работать и на процессорах Intel, и на ARM, что устроит и Intel, и Nokia.

То есть Intel разрабатывает ПО для «чужих» процессоров?
Именно. Подозреваю, что без особой охоты – но чего не сделаешь ради сотрудничества. Итак, Nokia будет применять свой инструментарий для создания интерфейса пользователя, Intel развивает свою систему управления пакетами, и обе компании получают «любимые» процессоры. Такое ощущение, что на технические решения повлияла большая политика.

Подождите, вы что-то сказали о системе управления пакетами?
Не будем забывать о том, что все это базируется на Linux, и установкой ПО управляет менеджер пакетов. В проекте Moblin используется *RPM*, а Maemo предпочитает *APT*. Выбрали один, и в MeeGo остался *RPM*.

Ох, нет!
Я тоже не поклонник *RPM*, поэтому считаю выбор странным. Но, в конце концов, менеджер пакетов скрыт от пользователя за фасадом хранилища приложений...

Прекрасно — значит, пользователи смогут загружать приложения из одного централизованного источника?
Было бы вполне логично предположить, что приложения будут храниться в одном месте, как у Apple, но...

Не увиливайте! Так единого хранилища не будет?
К сожалению, нет. Вместо организации центрального «склада» приложений, Intel будет использовать AppUp, который проходит бета-тестирование. Nokia сохранит за собой прежнее хранилище Ovi.

Путаница какая-то. Может быть, Android наведет здесь порядок?
Android – это сила, но он еще более фрагментирован, чем MeeGo. Там множество разнокалиберных устройств, и все с разным аппаратным и программным обеспечением. Кроме того, поддержка Android была недавно изъята из ядра Linux, поскольку никто не проявил заметного интереса к сопровождению его именно там.

Похоже, вся эта неразбериха выльется в конкурентную борьбу?
Сообщество свободного ПО работает, порождая массу проектов на SourceForge; часть из них со временем становятся знамениты и очень популярны. Мы не знаем, кто окажется сильнее, MeeGo или Android – по большому счету, оба льют воду на одну мельницу.

И какая роль во всем этом отводится Ubuntu Netbook Remix?
Думаю, Ubuntu и здесь прибегнет к испытанной тактике: подождет, пока другие сделают всю работу, соберет воедино, соорудит нечто сногсшибательное и выкрасит в коричневое. Это срабатывало не раз и, похоже, повторится – популярность Ubuntu Netbook Remix затмит и MeeGo, и Android.

Где можно узнать подробности о MeeGo?
Вполне предсказуемо, информация рассеяна по сайтам Moblin и Maemo: http://meego.com пока слишком скуден. Поэтому сначала направляйтесь на www.moblin.org и www.maemo.org. Если все пойдет по плану, первое ПО появится к началу лета, а первые устройства будут изготовлены к концу текущего года. LXF





По рецептам доктора Брауна

Д-р Крис Браун
Доктор обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

Добротное администрирование систем из причудливых заворотов кишок серверной.

Подвигам всегда есть место

Можно мне немного побрюзжать? Не так уж часто я это делаю. Я извел два вечера, пытаясь заставить работать микрофонный вход моего ноутбука в Ubuntu. Мне он был нужен для *Skype* и для учебных онлайн-курсов на www.spreed.com. Я гуглил и грепил, читал форумы и угрожал компьютеру физической расправой, и все без толку. Мне попались утилиты, предлагающие на выбор входы с именами «HDA Intel STAC92xx Analog OSS», «LFE» или «IEC958 D». Бросьте, парни, на моем ноутбуке всего один внешний аналоговый вход, его я и хочу выбрать. Почему все так сложно? Будь я новичком, я бы плюнул на Linux и ушел в Windows: там микрофон работает нормально. По иронии судьбы, встроенная в крышку ноутбука маленькая камера прекрасно работала безо всякой дополнительной настройки.

Руководство по Ubuntu

А теперь о более приятном. Я был рад увидеть в работе проект под названием Ubuntu Manual Project: его цель – «создать и внедрить документ, пригодный в качестве отправной точки для освоения Ubuntu начинающими пользователями». Проект начат студентом Бенджамином Хамфри [Benjamin Humphrey]. Ubuntu надеется поддерживать документ на нескольких языках, начиная с версии 10.04. Чтобы включиться в проект, зайдите на https://wiki.ubuntu.com/ubuntu-manual.

Я просто надеюсь наконец выяснить, как доконать этот клятый микрофон...


















Стать частью петафлопа

Проект BOINC Если ваш процессор сачкует, пожертвуйте его лишние циклы на благие дела.

Вы, видимо, слышали о SETI (Search for Extra-Terrestrial Intelligence – Поиск внеземного разума); этот проект использует пожертвованное компьютерное время. Проект Berkeley Open Infrastructure for Network Computing (BOINC, Открытая инфраструктура Университета Беркли для сетевых вычислений) обобщил эту идею и предоставляет машинное время для самых разных целей. Согласно их сайту, программа создана для передачи «излишков» машинного времени научным проектам вроде ClimatePrediction.net, SETI@home, Rosetta @home и World Community Grid. На данный момент проект предоставляет около 4,3 петафлопа вычислительной мощности. Даже не пытайтесь представить, сколько флопов в петафлопе – это более чем в два раза превышает показатель самого мощного суперкомпьютера в мире, Cray XT5 (на котором, кстати, стоит SUSE). BOINC подходит для проектов, расчленяемых на части. Части загружаются на пожертвованные компьютеры, которые обрабатывают их и возвращают результаты – возможно, дни спустя. Пакеты программ существуют для Red Hat и Debian, они есть во всех основных репозиториях. В Fedora выполните команду:

```
#yum install boinc-client boinc-manager
```

Затем для запуска основного клиента наберите:

```
# service boinc-client start
# chkconfig boinc-client on
```

Управляется *Boinc* через графический интерфейс с помощью *boincmgr*. Базовое представление выглядит малость неуклюже, но у «продвинутого» варианта стандартный вид Gnome. Можно вывести список проектов, в которых вы зарегистрированы, и изменить их; узнать, какими задачами сейчас занят ваш компьютер; проверить использование диска, просмотреть журнал активности и т. д. После регистрации проекта обо всем заботится *Boinc*, но вы можете ограничить объем предоставляемых вами ресурсов и запретить подключать компьютер к новым задачам, а также решить:

- в какое время дня предоставлять ресурсы компьютера;
- какую долю процессора вы намерены отдать;
- сколько дискового пространства выделить;
- можно ли использовать компьютер, если он работает на батарее.

› Ваш компьютер пережует полученные данные и вернет результат BOINC.

Проекты BOINC

Проекты BOINC простираются от глубоко человечных до вопиюще абстрактных. Вы можете поддержать исследования изменений моделей климата, оценку производительности адиабатических квантовых компьютеров, определение трехмерной формы белка, поиск пульсаров или продолжить охоту за внеземными цивилизациями. См. http://boinc.berkeley.edu/projects.php.

Быстрое руководство по RPM

Управление пакетами Пользуйтесь RPM как профи, не читая man-страницу.

RPM – сокращение от *Red Hat Package Manager* (менеджер пакетов *Red Hat*). Название относится и к формату файлов пакетов, и к программе для управления ими. Данный формат определен в Linux Standard Base и используется (естественно) в Red Hat Linux, в производных Red Hat, таких как CentOS и Fedora, и в некоторых конкурирующих дистрибутивах, включая SUSE. По современным стандартам, RPM – примитивное средство, и большинство пользователей, наверное, предпочтут ему более высокоуровневые утилиты на основе репозиториев, которые поддерживают автоматическое разрешение зависимостей – типа *Yum*, *apt-get*, *gpk-application* или *Synaptic*.

Тем не менее, RPM сохраняет свое значение как основной инструмент сопровождения базы данных установленных пакетов, а также собственно их развертывания и опроса. И не помешает знать, как управлять им напрямую. Как и многие утилиты командной строки в Linux, *rpm* страдает излишком количества опций, и в нашем кратком руководстве я покажу, как выполнить основные задачи без необходимости нырять в man. В таблице справа приведены некоторые стандартные задачи и способ их решения.

Чтобы...	Воспользуйтесь командой:
Вывести список всех установленных пакетов	**rpm -qa** Часто передается в grep, чтобы найти заданные пакеты: **rpm -qa | grep init**
Показать информацию об установленном пакете	**rpm -qi initscripts**
Показать файлы, входящие в состав установленного пакета	**rpm -ql initscripts**
Выяснить, из какого пакета файл	**rpm -qf /usr/sbin/sshd**
Найти зависимости («Что еще нужно для того, чтобы этот пакет работал?»)	**rpm -qR openssh-server**
Найти зависимые пакеты («Что сломается, если я удалю этот пакет?»)	**rpm --whatrequires initscripts** Одни зависимости выводятся как имена пакетов, другие – как имена файлов, часто библиотек
Вывести сведения о еще не установленном пакете	**rpm -qip bzip2-1.0.3-3.i386.rpm**
Установить пакет	**rpm -ivh bzip2-1.0.3-3.i386.rpm**
Удалить пакет	**rpm -e openssh-server**
Выяснить, изменились ли какие-то файлы пакета с момента его установки	**rpm --verify openssh-server**

RPM подробно

Большинство примеров в таблице очевидны; попробуйте их с какими-нибудь пакетами на ваш выбор, а на некоторые задачи взглянем более детально.

Во-первых, рассмотрим случай, когда при попытке установки пакета *RPM* не удается разрешить зависимость:

```
#rpm -i agg-2.4-2.1.i386.rpm
error: Failed dependencies:
            libSDL-1.2.so.0 is needed by agg-2.4-2.1.i386
```

Здесь возможности *RPM* иссякают. Хотя ясно, что имеет место нехватка зависимостей, *RPM* не пытается установить их и хоть сколько-нибудь помочь. Например, какой пакет нужно установить, чтобы получить отсутствующую *libSDL*?

У *RPM* есть полезная функция проверки состояния имеющихся в системе файлов и сравнения их с содержимым базы данных установленных пакетов. Это похоже на запуск *Tripwire* для проверки, не изменилось ли что-нибудь, чему меняться не следует. Проверить это можно для одного пакета:

```
# rpm --verify initscripts
```

Или для всех пакетов:

```
# rpm --verify --all
```

Вот фрагмент вывода:

```
# rpm --verify --all
..5....T. c /etc/inittab
S.5....T.   /etc/cron.d/smolt
S.5....T. c /etc/libuser.conf
.M....... /usr/lib/pm-utils/sleep.d
S.5....T. c /etc/ldap.conf
.M....G.. /var/log/gdm
```

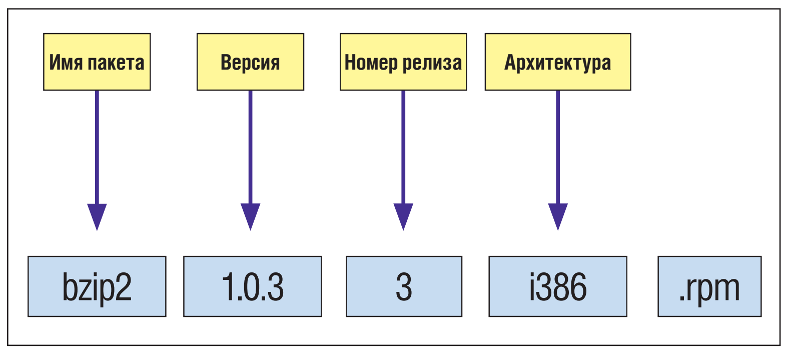
Для каждого файла, который не совпадает в точности с содержимым базы данных, RPM выводит различия в последовательности из девяти символов, включающих:

- **S** – отличается размер файла;
- **M** – отличаются права доступа к файлу;
- **5** – отличается контрольная сумма MD5;
- **D** – отличается младший/старший номер устройства;
- **U** – отличается владелец файла;
- **G** – отличается группа файла;
- **T** – отличается время изменения.

Так, мы видим, что, например, у файла **/etc/ldap.conf** размер, контрольная сумма и время изменения не те, что хранятся в базе данных *RPM*. Для конфигурационных файлов это нормально, и более половины рассогласований касаются именно их.

Именование имен

У пакетов RPM ужасные названия – невольно радуешься, что есть на свете дополнение имени файла. Но эти имена следуют определенному шаблону, показанному на рисунке справа. Обратите внимание, что фрагмент имен перед расширением **.rpm** означает архитектуру, для которой собран пакет. Архитектуры **i386** и **sparc** предполагают пакеты из скомпилированных двоичных файлов для соответствующего семейства процессоров. **Src** означает, что в пакете исходный код, а **noarch** сообщает, что пакет «не зависит от архитектуры». Такие пакеты обычно пишутся на интерпретируемых языках, например **Perl** или **Python**, или могут быть просто документацией.



Имена пакетов RPM-файлов содержат массу полезной информации, нужно лишь уметь ее прочесть. Здесь показано, что означает каждая секция.

Шлем верные сигналы

Сигналы Подробное руководство о том, что это, откуда они взялись, куда идут и что происходит при отправке.

Сигнал – событие, отправляемое выполняющейся программе ядром или пользовательским процессом. Сигналы асинхронны, то есть могут приходить когда угодно, и их появление в большинстве случаев никак не связано с тем, что собиралась делать программа. Сигналы обычно означают некое внешнее событие, способное повлиять на выполнение программы. Программа может выбрать, что должно произойти по приходу сигнала определенного типа. Она может игнорировать его (за одним важным исключением); запустить специальный обработчик сигнала и продолжить с того места, где остановилась; или просто выполнить «действие по умолчанию» для этого сигнала, что в большинстве случаев означает завершение процесса.

За исключением сигналов «реального времени», существует 31 тип сигнала. Каждый тип определяется именем (например, SIGTERM) и целым числом (например, 15). Получить список всех типов можно командой:

```
$ kill -l
```

Из этих 31 администратор должен знать, полагаю, шесть. Я свел их в таблицу.

Имя	№	Описание	Действие по умолчанию
SIGHUP	1	Инициирует перенастройку фонового сервиса	Завершить процесс
SIGINT	2	Ctrl+C с клавиатуры	Завершить процесс
SIGQUIT	3	Ctrl+\ с клавиатуры	Завершить процесс с дампом памяти
SIGKILL	9	Суперважный сигнал, который нельзя перехватить или игнорировать	Завершить процесс
SIGSEGV	11	Программа пыталась обратиться к памяти по неверному адресу	Завершить процесс с дампом памяти
SIGTERM	15	«Вежливая» просьба закончить работу	Завершить процесс

Сигнал SIGHUP обычно используется для предупреждения демона (это системный сервис, работающий в фоне), что конфигурационный файл изменен и его нужно перечитать. Многие демоны так и реагируют на SIGHUP. Примеры включают «суперсервер Интернета» *xinetd* и демон системного журнала *syslogd*. Демонам, поддерживающим SIGHUP, лучше отправить этот сигнал, чем перезапустить их – тогда существующие соединения с клиентами сохранятся, а при останове и перезапуске демона они будут разорваны.

«HUP» – сокращение от «hang up» (повесить трубку). Изначально этот сигнал генерировался драйвером терминала последовательного порта с подключенным на прием звонков модемом. Если телефонное соединение обрывалось, драйвер терминала замечал, что несущая исчезла, и отправлял SIGHUP оболочке, завершая ее. Идея состояла в том, чтобы гарантировать, что новый абонент не начнет случайно работать в оболочке предыдущего.

Сигнал SIGINT посылается активному процессу при нажатии Ctrl+C в его терминале. Его действие по умолчанию – завершить (убить) процесс. Поэтому «долгоиграющие» задания часто можно прервать, нажав Ctrl+C. Однако многие программы игнорируют или перехватывают SIGINT. Например, *less* при получении SIGINT не завершается, а запрашивает ввод команды.

Сигнал SIGQUIT посылается активному процессу при нажатии Ctrl+\ на клавиатуре. Он используется гораздо реже и менее известен, чем SIGINT. По умолчанию он также завершает процесс, но заодно заставляет оболочку создать файл **core**, содержащий образ памяти программы, для «посмертной» отладки. В большинстве дистрибутивов файлы **core** запрещены, потому что в настройке процесса **ulimit** (которая ограничивает использование ресурсов) максимальный размер файла **core** установлен в ноль. Чтобы проверить свои настройки, выполните команду **ulimit -a** и поищите встроенную команду **ulimit** на man-странице *bash*.

История о SIGSEGV

В первый раз я общался с Unix в так называемой «Edition 6". Я принес на работу копию, которую мне дал коллега в Открытом университете, и установил ее на наш PDP-11/60. На другой день я печатал некоторые man-страницы (другой печатной документации не было) и получил сообщение об «ошибке памяти». Решив, что компьютер неисправен, я выключил Unix и провел остаток дня за диагностикой оборудования, но никаких проблем не обнаружил. В конце концов я узнал, что «ошибка памяти» – это сообщение оболочки о том, что процесс завершился с SIGSEGV. Со временем я выяснил, что многие сообщения об ошибках в Unix были скорее причудами программиста, а не попыткой объяснить, что же на самом деле происходит. Моим любимым было «Это не пишущая машинка».

Смертоносные сигналы

Сигнал SIGTERM отправляется командой **kill** по умолчанию. Это вежливая просьба «пожалуйста, приберите за собой и завершитесь». При выключении системы процесс *init* отправляет сигналы SIGTERM всем выполняющимся процессам с просьбой завершиться. Если это не срабатывает, *init* применяет более суровые меры...

Сигнал SIGKILL – самый подлый из всех. Его нельзя перехватить или игнорировать, и его действие – прикончить процесс. Отправка SIGKILL должна быть крайней мерой, только в том случае, если более благородные SIGTERM и SIGINT не сработали. Разница между SIGTERM и SIGKILL – как между вежливой просьбой завершиться и выстрелом в голову. SIGKILL не дает процессу возможности прибраться. Перед перезапуском «убитых» программ, которые временно хранили данные на диске или находились посреди двухфазной фиксации транзакции, может потребоваться ручная расчистка.

Сигнал SIGSEGV (нарушение сегментации) отличается от прочих описанных здесь тем, что не приходит извне, а обусловлен самой программой. Он подается, когда программа пытается обратиться к адресу памяти вне отведенного ей диапазона. (Это легко сделать в C и C++, сославшись на неинициализированный указатель). Блок управления памятью видит эту попытку и тихо говорит ядру, которое отправляет SIGSEGV программе. Действие по умолчанию при получении этого сигнала – завершить программу. Если в программе нет ошибок, такого произойти не должно, но безупречных программ мало, и вы, наверное, видели, как программы падают из-за ошибки нарушения сегментации.

Сигналы SIGILL (неверная команда), SIGBUS (ошибка шины) и SIGFPE (исключение с плавающей точкой) генерируются по-

Сигналы и статус выхода

Если программа запускается из командной строки и завершается сигналом, в оболочке появится сообщение об этом. Например, при завершении программы сигналом SIGTERM оболочка выводит сообщение "Terminated". Для программы, убитой сигналом SIGKILL, выводится сообщение "Killed", а для программы, совершившей самоубийство попыткой обратиться по неверному адресу памяти – сообщение "Segmentation fault". По статусу выхода оболочка знает, что процесс завершен сигналом.

У программы, завершившейся нормально, статус выхода равен нулю. Программа, которая (добровольно) завершается с ошибкой, возвратит статус выхода в диапазоне 1–127. А программа, завершенная сигналом, возвратит статус выхода, равный 128 плюс номер соответствующего сигнала. Мы можем увидеть это, запустив заведомо кривую программу на C и просмотрев статус выхода путем вывода значения специальной переменной оболочки **$?**

```
$ ./forcesignal
Segmentation fault
$ echo $?
139
```

Здесь статус выхода 139 (128+11) сообщает нам о том, что программа завершилась по сигналу 11 (SIGSEGV), что мы преднамеренно и пытались спровоцировать.

добным образом, когда программа пытается сделать такое, чего не разрешает аппаратная часть; но распространены они гораздо меньше. SIGSEGV легко продемонстрировать. Следующая программа на C из двух строк генерирует этот сигнал в ответ на попытку записать число по адресу памяти 0:

```
main()
{ *((int*)0) = 0;}
```

Отправка сигналов

Все сигналы в конечном счете доставляются ядром, но исходят из различных источников, как показано на рисунке справа. Для отправки сигналов из командной строки используется команда **kill**, обычно таким образом:

```
$ kill -SIGINT 11434 11559
```

11434 и **11559** – идентификаторы процессов-приемников. Отправить сигналы можно только тем процессам, владельцем которых являетесь вы (если вы не root). Вместо имени сигнала допускается его номер; можно также не набирать **SIG** в начале имени, поэтому три следующих команды эквивалентны:

```
$ kill -SIGKILL 11434
$ kill -KILL 11434
$ kill -9 11434
```

Команда называется «kill», потому что во многих случаях ее результатом становится завершение процесса; но это не самое лучшее имя. «Throw» [выбросить] или «send-signal» [послать сигнал] было бы лучше.

Неудобство **kill** в том, что нужно знать числовой идентификатор процесса. Команда **pkill** позволяет задать процессы, которым отправляются сигналы, другими способами. Вот три примера:

```
$ pkill -SIGHUP syslogd
$ pkill -SIGTERM -U fred
$ pkill -SIGKILL -P 13579
```

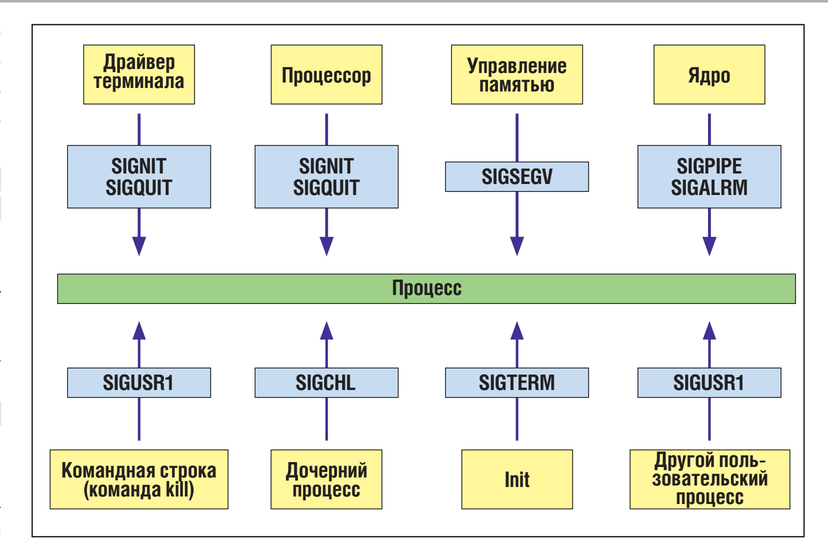
Первый сигнал отправляется всем процессам, выполняющим *syslogd*. Второй – всем процессам, владельцем которых является **fred**, а третий – всем процессам, чей родительский PID **13579**.

В оболочке есть встроенная команда **trap**, упрощающая обработку сигнала в сценариях без необходимости сочинять нудный код на C, как делают записные хакеры. **Trap** принимает два аргумента: команду (или набор команд) и имя сигнала, и обеспечивает выполнение указанных команд при получении сигнала с заданным номером. Попробуйте создать такой скрипт:

```
trap 'echo I got a SIGHUP signal' SIGHUP
trap 'echo Terminating on SIGTERM; exit 1' SIGTERM
while true
do
        sleep 1;
        date
done
```

Сделайте его исполняемым и запустите в фоне – так вы легко сможете узнать идентификатор процесса:

```
$ chmod u+x trapdemo
$ ./trapdemo &
[1] 18729
```



› **Сигналы могут возникать в разных местах, но все они передаются ядром.**

Скрипт печатает дату и ждет секунду. Поэкспериментируйте с командой **kill**, отправляя сигналы различных типов с другого терминала. При отправке SIGHUP скрипт сообщит, что получил сигнал, и продолжит работу. В ответ на SIGTERM он выведет сообщение и завершится. При получении любого другого сигнала выполнится действие по умолчанию.

Советы программистам

Программистам также могут быть интересны различные сигналы, но им нужно разбираться в сигналах слегка по иным причинам. Наверное, главное, что нужно знать программисту о сигналах – как заставить программу игнорировать их. Причин такого поведения может быть множество, но надо уметь обращаться с сигналами не только затем, чтобы просто наделить процессы возможностью выживания: сигналы обеспечивают обратную связь от системы.

Вот еще несколько вещей, для которых программисту могут пригодиться сигналы:

- » обработать SIGHUP для перечитывания конфигурационных файлов, обычно для демонов;
- » обработать SIGTERM, чтобы убрать за собой и корректно завершить работу;
- » обработать SIGCHLD, чтобы асинхронно собрать результаты завершения дочерних процессов;
- » обработать SIGALRM для реализации таймаута при блокирующих операциях;
- » обработать SIGUSR1 (или другой сигнал по вашему выбору), чтобы сообщить текущий статус процесса;
- » использовать сигнал, чтобы динамически включать и выключать отладочные сообщения.

Если вы серьезно заинтересовались сигналами с точки зрения программиста, раздобудьте книгу Ричарда Стивенса «UNIX. Профессиональное программирование» и погрузитесь в ее десятую главу. LXF

LINUX FORMAT Учебники

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Выжить рядом с РАО

Глупости какие-то говорите… Но это не страшно. Страшно то, что вы их ещё и думаете….
Вера Игоревна Илясова

РАО не даёт окружающему миру соскучиться. С лёгкой руки этой организации, информационные поводы множатся, как грибы после дождя. «Наезды» на детей, танцующих под защищённую музыку на открытом конкурсе, и ветеранов, поющих защищённые песни на празднике, восхищают: вполне по букве закона! При этом в РАО считают себя обиженными и непонятыми.

На общество им скорее всего наплевать, но начальство – статья другая. Чтобы сделать своё дело более понятным высшему руководству, самому Путину (интересно, почему не Медведеву?) было отправлено письмо с просьбой защитить бедное РАО от злобных завистников. Его подписали более чем 300 «деятелей искусства».

Имеют право. Законов формально не нарушают, а совесть сегодня не ходкий товар. Я вовсе не против вознаграждения авторам – я всей душой «за», но жадность РАО настораживает.

По умолчанию клиентами РАО являются абсолютно все, даже если они этого не жаждали. Чтобы отказаться от этого, нужно как-то известить данную организацию. Такое возможно – на сайте РАО имеется список «отказников»; неясен только алгоритм действий. Достаточно ли послать в РАО e-mail, требуя внести свою фамилию в данный список, или необходимо лично явиться с паспортом?

Придет время, когда отказ от навязываемых РАО услуг станет обязательным признаком порядочности.
E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Украшать рабочий стол 56
Время весенней уборки! Разгребите рабочий стол и украсьте его вместе с **Энди Ченнелом**. Плюс: родительский контроль.



Смотреть ТВ по-новому 60
Кабель и спутник – вчерашний день: **Марк Нассел** объясняет, почему *Boxee* – правильный способ смотреть передачи как и когда вы хотите.



Играть в классику DOS 64
Заела ностальгия по старым добрым «досовским» играм? **Адитья Шеваде** запустит их для вас в *DOSBox* на обычной машине.



Искать причины 68
R, увы, не скажет вам, в чем смысл жизни – но с его помощью **Сергей Петров** и другие прольют свет на практику принятия кадровых решений.



Управлять временем 72
Не позволяйте проектам выходить за рамки бюджета и сроков – послушайте мудрого **Шашанка Шарму** и переложите работу на *TaskJuggler*.



Кодировать с Clutter 76
Clutter – превосходный инструментарий, но и у него есть предел. Когда он достигается, **Ник Вейч** зовет на помощь *Cairo*.



Резервировать данные 80
Юрий Винник призывает *PING* и *BackupPC*, чтобы спасти и образы операционных систем, и пользовательские документы.



Защищать данные 84
А вот вам альтернативный подход: **Бен Мартин** познакомит вас с RAID-5. Пользуйтесь вместе, а не вместо!

Совет месяца: Удаленная оболочка



Screen весьма уместен при работе через SSH, ведь он позволяет отключаться от сервера и подключаться вновь, не прерывая работы задач. Эта идея станет ясна, если вы запустите через SSH длительную операцию, не используя *screen*, и вдруг осознаете, что надо бы закрыть соединение. Чтобы такое не повторилось, добавьте в профиль (**~/.bash_profile** или **~/.bash_login** – лично для себя, или **/etc/profile**, на благо всем) строку

```
if [[ -n “${SSH_TTY}” ]] && [[ -z “${STY}” ]]; then
screen -xRR
fi
```

Теперь, соединяясь с удаленным компьютером через SSH, вы автоматически подключитесь к сеансу *screen* или откроете новый, если *screen* еще не выполняется. Код проверяет наличие переменной окружения **$SSH_TTY** (признака подключения через SSH) и отсутствие **$STY**, означающей, что вы уже в сеансе *screen* (иначе ваш профиль запустил бы *screen*, который прочел бы этот файл и запустил *screen* повторно). Опция **-xRR** велит *screen* подключаться к действующей сессии в многопользовательском режиме: сеанс, активный на другом терминале, не будет принудительно прерван. Это удобно, если *screen* запущен на одном компьютере, а вы решили подключиться с другого.

Выход из *screen* по Ctrl+D или Ctrl+A D вернет вас в оболочку SSH. А если требуется, чтобы при выходе из *screen* оболочка завершала работу, примените

```
screen -xRR && exit
```

На вид и вкус:

Ваш компьютер и вы сами заслуживаете красивого рабочего стола. **Энди Ченнел** пройдется по настройкам, доступным пользователям KDE и Gnome.





Наш эксперт

Энди Ченнел
Энди, видимо, будет делать Первые шаги в Linux вечно, а технологиями он заинтересовался, открыв для себя Dragon 32.

Разработчики вложили много сил в обеспечение возможности изменить практически любой элемент рабочего стола Linux. Более того, дизайнеры и фотографы по всему миру трудятся над изготовлением качественных изображений для украшения вашего экрана. Жаль, если они пропадут! На данном уроке мы рассмотрим инструменты Gnome и KDE для настройки внешнего вида вашего рабочего стола – от новых иконок и обоев до хранителей экрана и шрифтов.

Первым делом изменим заставку рабочего стола. Для получения наилучшего результата необходимо узнать разрешение вашего монитора; простейший способ выяснить его – открыть терминал и вызвать *xdpyinfo*.

```
$ xdpyinfo | grep 'dimensions:'
```

Первая часть выполняет команду xdpyinfo; *grep* – это утилита, производящая поиск в файлах и отображающая результаты; а последняя часть определяет, какую строку искать. Вы увидете два числа. Первое из них – количество пикселей в вашем мониторе по горизонтали (в нашем случае это 1440), а второе – вертикальное разрешение (у нас – 900); теперь вы можете задать поиск изображений (эффектных или, наоборот, спокойных) с этими параметрами.

Большинство сайтов с обоями, включая www.gnome-look.org и www.kde-look.org, разбивают свои изображения по разрешению, и подходящий размер всегда найдется. После выбора и загрузки изображения в некое запоминающееся место на вашей машине можно щелкнуть на рабочем столе правой кнопкой мыши и выбрать Change Desktop Background [Сменить фоновое изображение] (Gnome) или Desktop Settings [Настроить рабочий стол] (KDE).



› Gnome поставляется с набором готовых тем, а еще больше можно накопать в Интернете.

В Gnome вы увидите окно Appearance Preferences [Настройки отображения], с набором обоев, которые можно установить. Выберите одно из них или нажмите кнопку Add [Добавить], чтобы перейти туда, где находится сохраненное вами изображение. Фон изменится, как только вы выберите или добавите изображение.

Внизу окна имеется удобная ссылка 'get more' [дополнительно]; кроме того, можно просто перетащить картинки в окно из другого места. Если размер или форма изображения не годятся для вашего рабочего стола, изучите настройки внизу (Zoom [Масштаб], Title [Заголовок] и т.д.).

В KDE все немного сложнее, но можно изменить заставку рабочего стола – а не всю тему – выбрав в выпадающем списке Picture [Изображения] или щелкнув на иконке Folder [Каталог] для поиска сохраненных изображений. И вновь, имеется несколько опций для масштабирования и заполнения, а также есть кнопка Get New Wallpapers [Загрузить новые обои], чтобы найти их в сети.

Помимо изменения фона рабочего стола, в обоих этих местах имеются опции для изменения или определения новых тем, включая фон, оформление окна, цвета и шрифты. Все они регулируются по отдельности.

Способ KDE

Чтобы добраться до инструментов определения внешнего вида и поведения KDE, откройте K-меню и выберите Computer > System Settings [Компьютер > Параметры системы] для вызова панели управления. Найдите раздел Look and Feel [Внешний вид и поведение]; в нем имеется подраздел Desktop [Рабочий стол], позволяющий настроить все эффекты рабочего стола. Пока мы пропустим его и выберем Appearance [Внешний вид]. Откроется новое окно со списком подразделов слева и основной панелью настройки, меняющейся в зависимости от выбранного, справа. Начните

» **Месяц назад** Мы изучали жуткую командную строку, пользователей и группы.

Рабочий стол

с пункта Style [Стиль]. В этом разделе две вкладки; первая, тоже озаглавленная Style, позволяет выбрать одну из готовых тем для KDE, пользуясь выпадающим списком Widget Style [Стиль графических элементов]. В центре окна вы увидите предпросмотр текущего стиля – нажатие кнопки Apply [Применить] применяет его ко всей системе. Добавочные опции доступны на вкладке Fine Tuning [Тонкая настройка]; среди них – размещение текста и иконок, одинаковые для всех тем, и кнопка Configure [Настроить], в меню которой находятся инструменты для изменения ширины полосы прокрутки, формы стрелочек в древе файлов и другие элементы. Доступные здесь опции зависят от выбранной темы.

Раздел Colours [Цвета] позволяет задать цвет практически любого элемента интерфейса, от рамки окна и его заголовка до фона панелей инструментов и выделенного текста. Это всеобъемлющий подход, но имеются и готовые цветовые схемы, которые можно применить ко всему рабочему столу.

Раздел Шрифты представляет перечень мест типа Toolbar [Панель инструментов], Menu [Меню] или Desktop [Рабочий стол], для которых можно указать шрифт, просто щелкнув на кнопке Choose [Выбрать] и отметив желаемый. Вам покажут все шрифты, имеющиеся в системе, поэтому выбирайте осторожно. Помните, главное – чтобы вам работалось проще и быстрее; использование вычурного начертания, очевидно, не лучшая идея. Выбрав гарнитуру, можно настроить некоторые параметры сглаживания, чтобы шрифт на экране выглядел лучше. Используйте System Settings [Системная настройка] или выберите Enabled [Разрешить] для активации дополнительных параметров в Configuration [Настройки].

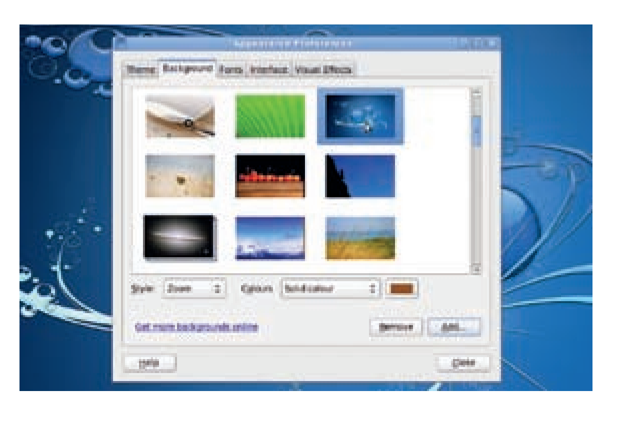
Последний интересующий нас раздел – Windows [Окна], в нем определяется вид границ окна. Присутствует выбор тем, а также инструменты для изменения выравнивания текста и добавления теней; взглянув же на вкладку Buttons [Кнопки], вы сможете изменить расположение кнопок Minimize [Свернуть], Maximize [Развернуть] и Close [Закрыть]. После всех изменений не забывайте нажать кнопку Apply [Применить], чтобы они вступили в силу.

Способ Gnome

Gnome имеет схожий с KDE набор опций, но общий подход проще: большинство настроек можно найти в System > Preferences > Appearance [Система > Настройки > Внешний вид], и вместо расположенных сбоку опций имеется набор вкладок.



› Опции настройки KDE выглядят сложнее, но выполняют те же функции, что и их аналоги в Gnome.



› Gnome поставляется с набором обоев, но еще больше можно добавить из различных источников.

На первой, Theme [Тема], выполняются всеобъемлющие изменения внешнего вида и поведения, и предлагается выбор уже готовых тем. Щелкните на теме или выберите Get More Themes Online [Загрузить дополнительные темы] для открытия вашего браузера и загрузки новых.

Для углубления в сами темы, выберите опцию Customise [Изменить] у нижнего края окна. Это откроет диалог для настройки каждой части интерфейса, включая Controls [Элементы управления] (ну, для нас с вами это виджеты), цвета, границы окна, иконки и указатели мыши. Выбор любой вкладки отобразит все имеющиеся опции. Дополнительные настройки появляются с добавлением каждой новой темы.

Определив тему в этом разделе, можете нажать кнопку Close [Закрыть], чтобы вернуться в основной интерфейс, а затем Save as [Сохранить как] для сохранения файла темы для последующего использования: это очень удобно, если вы сторонник реформ интерфейса.

Интерфейс Fonts [Шрифты] позволяет определять гарнитуру, используемую для различных элементов интерфейса, таких как заголовок окна и рабочий стол. У нижнего края этого диалога находится раздел рендеринга из четырех частей, позволяющий определить, как шрифты отображаются на мониторе. Если у вас современный плоский монитор, то лучшей опцией будет Subpixel smoothing (LCDs) [Субпиксельное сглаживание (ЖК)], которое сгладит шрифты, убрав ступенчатые края.

Во вкладке Interface [Интерфейс] настраивается отображение иконок и соответствующего им текста, а также включение пиктограмм в меню. Предпросмотр показывает влияние ваших изменений.

Наконец, вкладка Visual Effects [Эффекты] позволяет настроить широкий диапазон параметров того, как компьютер будет управлять украшениями, предоставляемыми *Compiz*. Доступны три опции. None [Отключено] – для малопроизводительных компьютеров (или тех, кто не любит блескучую анимацию); Normal [Обычный] включает некоторые эффекты вроде прозрачности окон при их перетаскивании; Extra [Полностью] включает все, от скользящих виртуальных рабочих столов и резиновых окон до эффекта Аладдин и прозрачности меню (их использование мы рассмотрим подробнее в следующий раз), но для его применения вам нужна действительно хорошая видеокарта.

В отличие от KDE, настройки Gnome обновляются в реальном времени, поэтому в диалогах нет кнопки Apply [Применить] – все происходит сразу при выборе.

Скорая помощь

Если вы пользуетесь поиском изображений в Google, то перейдите в Advanced Search [Расширенный поиск] и найдите опцию для задания размера, под названием Use my desktop size [Разрешение моего рабочего стола].

Первые шаги под

Хотите защитить своих детей в Интернете? Мы покажем как.

Интернет – это целая вселенная классных приложений, поучительных сервисов и захватывающей информации. Но она содержит и такое, чего вы не желали бы показывать своим детям; и имеется несколько способов держать их подальше от определенных мест или ограничить их доступ перечнем сайтов, в безопасности которых вы уверены. В совокупности с управлением пользователями, которое мы обсуждали в прошлом номере, это поможет вам создать полную систему родительского контроля вашего компьютера.

Для контроля за доступом в Интернет, мы установим и настроим два различных дополнения *Firefox*. Первое позволит задать набор правил, таких как ключевые слова и фильтр сайтов, а второе создаст полностью «дружелюбное к ребенку» окружение, защищенное от наиболее опасных или вульгарных частей Интернета.

Glubble и ProCon

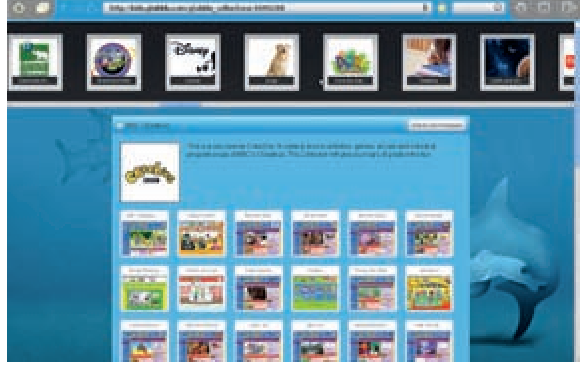
Сперва мы установим модуль расширения под названием *Glubble*. Получить приложение вы можете на www.glubble.com по ссылке Download [Загрузить]. Это стандартный файл **.xpi**, но поскольку мы берем его не с сервера Mozilla, вам может потребоваться авторизовать загрузку, щелкнув на кнопке Allow [Разрешить], которая появится перед установкой в верхнем правом углу вашего экрана. Затем следует перезапустить браузер.

При первом старте следует зарегистрироваться в службе, или войти с имеющимися именем и паролем вашей учетной записи Google, Yahoo, Facebook или MS Live. Служба добавит новую панель инструментов в *Firefox*; щелкните на кнопке Child Mode [Режим ребенка], блокирующей браузер (для выхода из режима ребенка потребуется ввести специальный код). Браузер закроется, а затем запустится вновь, в защищенном режиме.

В качестве второго, менее драконовского варианта, мы выбрали модуль *ProCon Latte*. Он не работает со всем браузером, а пытается фильтровать сайты по определенным словам (можно добавлять свои) и темам, а также может использоваться для настройки «белого списка» доступных сайтов. Первое, что необходимо выполнить после установки (обычным образом) – это определить пароль на доступ. Сделайте это, открыв модуль (Tools > ProCon [Сервис > ProCon]) и выбрав опцию Пароль [Password] у нижнего края окна. Пароль будет запрошен дважды. Теперь можно пробежаться по списку доступных опций и включить или выключить их. Все начинается с первого пункта, а именно фильтра Enable Explicit Material [Фильтровать откровенные материалы] на вкладке Main Filter [Основной фильтр]. Включите ее, и все web-страницы начнут фильтроваться. Если ваши непоседы заблудятся в джунглях Интернета – или примутся искать их – в верхней части экрана появится предупреждающее сообщение о том, что сайт был отфильтрован.

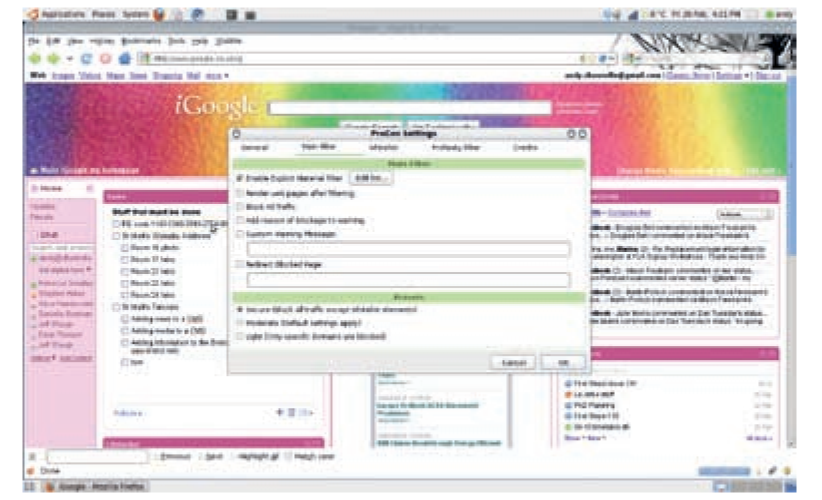
Используйте опции внизу экрана для настройки уровня фильтрации. Он может меняться от Strong [Сильный], когда отображаются только сайты из белого списка (который необходимо создать самостоятельно), до среднего и слабого. Чтобы облегчить составление списков, можно включить опцию Add to Whitelist [Добавить в белый список] в контекстное меню – это делается на вкладке General [Общие]. Теперь при перемещении в сети вы можете щелкнуть на любой странице правой кнопкой мыши и, выбрав пункт в меню *ProCon*, или блокировать данный сайт (то есть добавить в черный список), или разрешить доступ к нему при запущенном фильтре, добавив его в белый список.

Вы можете определить, запущен ли фильтр, а также включить или выключить его, используя небольшую иконку щита в правом нижнем углу окна браузера.



› *Glubble* **предоставляет в своем окне обширный список безопасных сайтов — включая массу образовательных.**

› Процесс прививания навыков безопасного серфинга в вашей семье начинается здесь.



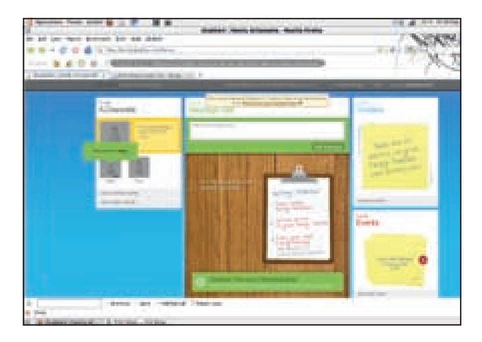
Помните об ответственности

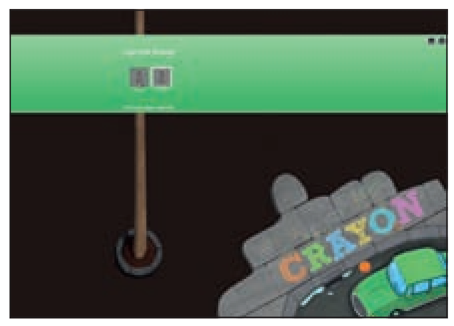
Ни фильтры, ни белые списки не идеальны; и хотя они и могут служить первой линией защиты детей от всевозможной грязи из сети, вы должны всегда знать, чем они там занимаются. Как заядлый компьютерщик, вы должны уметь просматривать историю в *Firefox* (История > Показать всю историю) и управлять своей собственной при помощи Правка > Настройки > Настройки безопасности.

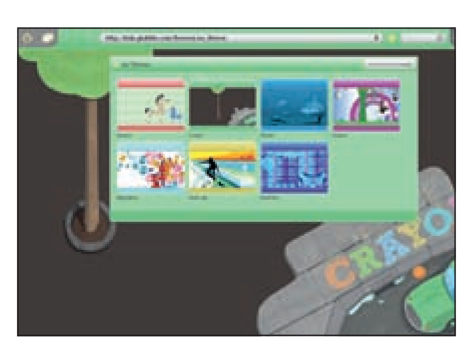
опекой родителей



Шаг за шагом: Настроим безопасный серфинг







1 Добавим ребенка в Gubble

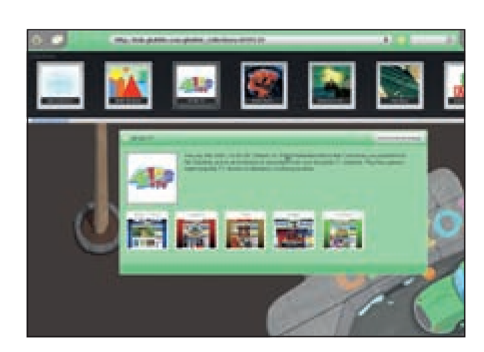
На панели инструментов, перейдите на страницу Family [Семья] и авторизуйтесь. В левой части страницы вы увидите вашу собственную иконку, а под ней ссылки для добавления к сервису взрослого [adult] или ребенка [child]. Выберите последнее. Потребуется ввести имена и даты рождения детей, а можно и добавить фото.

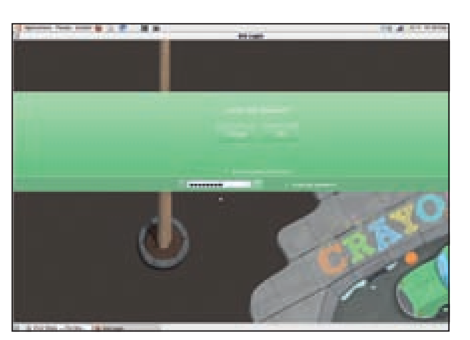
2 Позволим ему войти

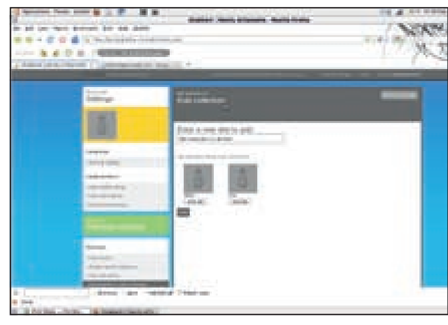
Настроив детские учетные записи, можно щелкнуть на иконке Child Mode [Детский режим] (вторая слева) для блокирования браузера. *Firefox* закроется и загрузится вновь в детском режиме. Ваши малыши выберут свое имя или фото в окне входа и окажутся перед собственным домашним экраном.

3 Настроим пространство

Каждый ребенок может настроить окружение под себя, щелкнув по иконке тем (выглядит как винтик) и выбрав одну из имеющихся. На домашней странице они могут писать короткие сообщения (которые будут появляться в панели инструментов) и создавать коллекции любимых сайтов при помощи звездочки в адресной строке.







4 Ищем в избранном

Система дает доступ ко множеству сайтов. Нажмите на кнопку Collections [Коллекции] (слева вверху), а затем прокрутите верхнюю панель в поисках определенного сайта. Выбор одного из них приведет к отображению доступных страниц в главном окне, а щелчок на них перенесет вас к ресурсу. Все содержимое страницы отображается, но внешние ссылки могут не работать, если только они не ведут к сайтам из белого списка.

5 Вернемся

Чтобы вернуться в стандартный *Firefox*, нажмите кнопку Close [Закрыть] справа вверху экрана, а затем, в окне входа, нажмите кнопку Parent Unlock [Разблокировать для родителей]. Будет запрошен пароль, а затем вы вернетесь обратно в нормальный браузер. Он даже вспомнит все сайты и вкладки, которые были открыты.

6 Добавим в список свой сайт

В адресной строке обычного браузера вы увидите иконку. Перейдите на «безопасный» сайт и нажмите ее. Произойдет перенаправление к окну *Gubble*, в котором вы сможете определить, кому из ваших детей разрешается видеть данный ресурс. При следующем их входе он будет доступен в их коллекции. LXF

» **Через месяц** Эффекты рабочего стола и интерактивные презентации в *OOo*.

Boxee: Больше

Марк Нассел раскрывает «что» и «почему» *Boxee* – инструмента медиа-вещания для поколения социальных сетей.





Наш эксперт

Марк Нассел менеджер технических проектов с более чем 20-летним опытом работы в индустрии ИТ и телекоммуникаций. Пишет о Linux и Unix в различных изданиях с 1997 года.

B*oxee* – кросс-платформенный медиа-центр, на данный момент доступный для Ubuntu Linux, Mac OS X, MS Windows и Apple TV. Его основное назначение (как и других подобных продуктов) – объединение различных форматов мультимедиа в единое пользовательское окружение. Для *Boxee* эти форматы включают потоковое интернет-аудио и видео, подкасты, локальные аудио- и видеофайлы, DVD и CD. Он не предоставляет возможностей записи звука или видео, в отличие от своих конкурентов *MythTV* или *Windows Media Centre*, зато реализует другой подход, сплавляя мультимедиа с социальными сетями.

Boxee – не просто цифровой видеомагнитофон: он делает упор на социальном аспекте использования медиа. *Boxee* пытается впрячь коллективный разум, интегрируя в окружение элементы социальной сети и рекомендации по контенту. Пользователи оставляют оценки просмотренного видео и рекомендации, которые обобществляются несколькими способами. Во-первых, можно размещать комментарии в популярных социальных медиа-сервисах, типа Twitter и Facebook, нажатием кнопки пульта или кнопкой мыши. Формат сообщения по умолчанию таков:

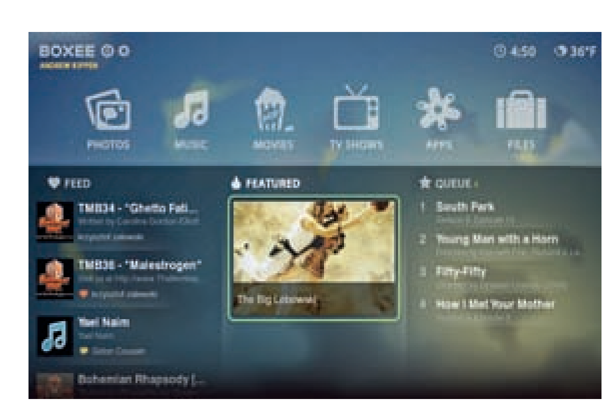
```
<username> Likes <program_name> on Boxee <url>.
```

[<пользователю> понравилась <программа> на Boxee <адрес>]. Если вы хотите изменить его, у вас есть возможность добавить собственный текст.

Во-вторых, платформа отслеживает ваши рекомендации и разделяет их с вашими друзьями в *Boxee*. *Boxee* поддерживает концепцию друзей или последователей, аналогично Twitter или FriendFeed. Пользователи могут просматривать рекомендации друзей, зайдя на сайт www.Boxee.tv или в столбце **Feed [Лента]** приложения (по рекомендациям друзей я нашел немало отличных программ).

Кроме социальных медиа-функций, *Boxee* предлагает несколько других возможностей. Простота поиска содержимого – ключевое свойство медиа-центра. В современном мире облачных вычислений средства поиска выходят за рамки локальных дисков и DVD и прорываются в Интернет. *Boxee* предоставляет широкий поисковый функционал, помогающий пользователям найти нужные программы. Прекрасный пример – поиск телешоу (TV Show search), который ищет программы не только на локальных и сетевых дисках, но и в Интернете. Для найденных программ *Boxee* отображает места, откуда можно их загрузить. По щелчку на одном из источников начнется воспроизведение программы. Эта функция реально упрощает поиск любимых программ.

Другой ценный компонент *Boxee* – приложения (app): они расширяют стандартные возможности медиа-сервера RSS-лентами и прямым доступом к источникам. Приложения *Boxee* пишутся на Python и XML, и многие из них поставляются сообществом. Прекрасный пример – Clicker App. Приложение помогает пользователям при поиске фильмов, видео и ТВ-программ через Интернет. Результаты отображаются в виде «черепицы» с функцией сортировки и фильтрации. По щелчку на клетке выводится информация о программе и источнике. Потом можно воспроизвести фильм или вернуться к списку. Пользователи также могут просматривать содержимое по названию [Show], источнику



› **Проглядите, что смотрят ваши друзья, рекомендации персонала *Boxee* и список воспроизведения — все на одном экране.**

Поддерживаемые форматы

Boxee поддерживает достаточно большое число форматов:

» AVI	» MP4 (H.264)	» WMA
» MOV	» AAC	» BMP
» FLV	» H.264	» JPEG
» FLI	» AIFF	» GIF
» MPG	» WAV	» PNG
» MP4	» RealAudio	» TIFF

Распознаются также DVD и CD, но не файлы, защищенные DRM.

телевидения

[Sources] или категории [Categories]. На данный момент в системе около 160 приложений, и постоянно появляются новые.

Вместо текста в интерфейсе *Boxee* широко используются иконки и картинки, чтобы облегчить просмотр приложения на различных экранах, особенно телевизионных. По этой причине к интерфейсу было применено «правило 10 футов» (оно гарантирует, что все элементы управления и важный текст видны с расстояния 10 футов [около 3 м, – *прим. пер.*] на обычном экране). Поэтому *Boxee* одинаково хорошо выглядит и на маленьких экранах нетбуков, и на новых плазменных телевизорах. *Boxee* реализует кросс-платформенное видеовещание. Изрядная доля телезрителей недовольна программной или ценовой политикой традиционных источников видеоконтента – *Boxee* призван учесть их интересы, вернув управление в руки зрителей.

Своими корнями *Boxee* уходит в XBMC, медиа-центр с открытым исходным кодом. В 2007 году часть проекта ответвилась в *Boxee*, добавив интернет-потоки и социальные сети. *Boxee* оставался в альфа-статусе до января 2010 года, когда была выпущена новая открытая бета-версия. Бета-версия *Boxee* – серьезный шаг вперед по сравнению с исходным проектом. Были исправлены многие давние ошибки, программа стала значительно удобнее, взлетела производительность.

Начинаем

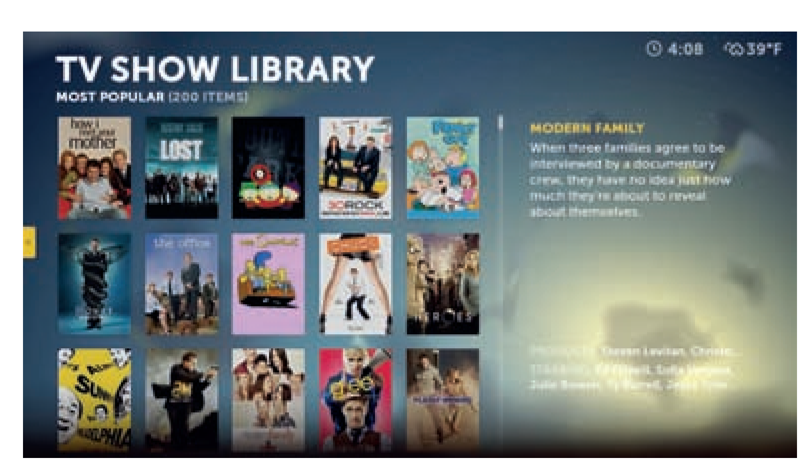
В Ubuntu 9.04 Jaunty Jackalope и 9.10 Karmic Koala установка довольно проста – загрузить пакеты можно с сайта *Boxee*. Те, кто любит приключения или не любит Ubuntu, могут скачать исходный код. На нашем уроке мы предполагаем, что для установки используется официальный пакет Ubuntu. Так как продукт еще в бета-версии, загрузите пакеты прямо с сайта *Boxee*. В феврале этого года было выпущено существенное обновление бета-версии *Boxee*, исправившее более 500 ошибок. С появлением новой версии приложение не обновляется автоматически, и пользователям нужно периодически проверять сайт.

Официальные аппаратные требования *Boxee* по современным меркам мизерны:

» x86-совместимый процессор Intel/AMD с тактовой частотой 1 ГГц или выше.
» 512 МБ оперативной памяти или больше.
» Графический контроллер/карта/адаптер и драйверы устройства, совместимые с:
  » Mesa OpenGL 1.4 (+ GLSL).
  » Direct X 9.0.
  » 32-битный цвет или более.

Boxee будет комфортно работать даже на нетбуках с процессорами Atom. Большинство младших моделей Asus Eee PC, HP Mini и Gateway поладит с ним за милую душу. Главное, чтобы видеокарта соответствовала минимальным требованиям.

Одна из более интересных возможностей *Boxee* – социальные сети. Пользователи могут следить за действиями друзей и делиться оценками и советами по просмотру. Если у вас есть учетная запись в Twitter, Facebook, FriendFeed, Tumbler, Netflix, Flickr или Digg, можно разместить свои оценки и рекомендации, не покидая *Boxee*. Чтобы воспользоваться большей частью этого функционала, начните с создания учетной записи пользователя на сайте www.Boxee.tv. Покончив с этим, зайдите на вкладку Settings [Настройки] и выберите сервис, к которому хотите подключиться, а также информацию, которую хотите передать.



› **Ищите свои любимые программы в Интернете, на локальных дисках или других домашних компьютерах.**

Обзаведясь учетной записью, загрузите самый свежий установочный пакет для 32- или 64-битной версии, в зависимости от вашей операционной системы. По установке пакета в меню Sound and Video [Звук и видео] автоматически появится иконка.

Интерфейс Boxee

По умолчанию *Boxee* открывается в полноэкранном режиме, но размер окна можно изменить, нажав клавишу обратного слэша (\). В верхней части экрана находится серия иконок меню: Photos [Фотографии], Music [Музыка], Movies [Фильмы], TV Shows [ТВ-программы], Apps [Приложения] и Files [Файлы]. *Boxee* – полноценный медиа-центр, и может обрабатывать файлы большинства распространенных мультимедиа-форматов.

По щелчку на фотографиях, музыке или фильмах отображается связанное содержимое, хранимое на вашем компьютере. По умолчанию *Boxee* использует для хранения мультимедиа-файлов пути **домашний_каталог/Pictures**, **Music** и **Movies**. Если файлы хранятся в других местах, путь легко изменить »

Скорая помощь

Чтобы изменить настройки видео во время просмотра, нажмите Enter, кликните по иконке Settings [Настройки] в виде шестеренки, затем на Video [Видео].

Хула Hulu

Hulu – очень популярный сайт потокового видеоконтента, поддерживаемый совместно NBC, Fox и ABC. На нем пользователи могут смотреть новые и старые выпуски программ. Концепция похожа на кабельное телевидение по запросу: для просмотра программ необходим лишь браузер с поддержкой Flash. Однако, когда *Boxee* начал использовать Hulu как один из источников контента, ребята из Hulu разобиделись, и разгорелись публичные споры об использовании контента в *Boxee*.

Противостояние между Hulu и *Boxee* пришло к финалу весной 2009 года, когда Hulu заблокировал доступ *Boxee* к своему содержимому. При каждом изменении сайта Hulu разработчики *Boxee* снова обходили защиту. *Boxee* даже завел учетную запись в Twitter, чтобы его почитатели знали о статусе подключения.

Решением *Boxee* было получать доступ к контенту Hulu через браузер внутри приложения, точно так же, как обычный пользователь делает это в *Firefox*.

» **Пропустили номер?** Узнайте на с. 107, как получить его прямо сейчас.



Чтобы изменить порядок иконок ярлыков во всплывающем меню, нажмите кнопку Escape на главном экране (Home screen), перейдите вниз к пиктограммам и выберите иконку с шестеренкой.

в Настройках [Settings]. Иконка телепрограмм позволяет искать последние в Интернете, на сетевых дисках и локальных файловых системах. Приложения представляют собой настроенные ленты и подключения к другим сайтам. Группы программ, такие как подкасты Revision3, новостные репортажи CNN или удивительное видео Cool Hunting, доступны за один щелчок. У приложений обычно есть свой собственный список программ и меню. Последняя, но не по значимости – иконка Files [Файлы]. *Boxee* позволяет обращаться к файлам в локальных файловых системах и в сетевых каталогах. В разделе Файлы можно искать разделяемые ресурсы и/или создавать ярлыки. Если монтируется DVD или CD, связанная с ним иконка предоставит доступ к устройству. *Boxee* может воспроизводить DVD-видео и компакт-диски точно так же, как *VLC* и другие видеоплейеры.

Область под меню иконок разделена на три части: первая – Feed [Ленты] – содержит источники контента, рекомендуемые другими пользователями, за которыми вы следите. По щелчку на ссылке контент откроется в подходящей программе для просмотра. Вторая часть, Featured [Рекомендуем], содержит контент, рекомендованный командой *Boxee*. В последнем столбце отображается очередь просмотра. При поиске по фильмам и ТВ-программам результаты можно помечать для последующего просмотра (они будут попадать в эту очередь).

Чтобы изменить настройки, щелкните по иконке в форме шестеренки справа от логотипа *Boxee*. В разделе Настройки [Settings] можно задать параметры видео, звука, сети и профиля пользователя.

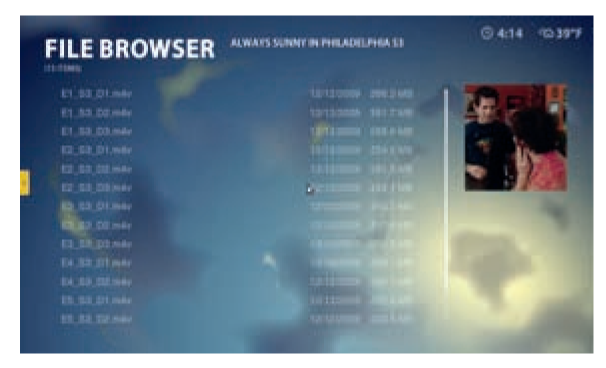
Смотрите, чего душа просит

В *Boxee* можно смотреть любимые подкасты, видеоролики YouTube, фильмы и телепрограммы в потоковом режиме, учебные видеокурсы, видео с Hulu, Netflix, DVD-диски, локальные видеофайлы и видео в общих каталогах локальной сети. Если вы меломан, можно пустить поток любимой музыки по дому или послушать интернет-радио. И не будем забывать о прекрасных фотографиях, которые вы сделали в отпуске – с помощью *Boxee* их можно посмотреть по телевизору. Потребуется только небольшой настольный компьютер или ноутбук, подключенный к Интернету и телевизору с помощью кабеля HDMI. Загрузите Ubuntu Linux, установите *Boxee* и настройте общий доступ к файлам на других компьютерах (мы поговорим о том, как это сделать, чуть позже).

По умолчанию Ubuntu не поддерживает воспроизведение зашифрованных коммерческих DVD-дисков – можно смотреть только домашнее видео. Но большинство людей хотят смотреть

Таблица региональных кодов

Код региона	Регион
0	Неофициальный параметр [неформальное обозначение «во всем мире»; диски с таким кодом региона либо не имеют определенного кода, либо имеют код 1–6, – *прим. пер.*]
1	Канада, США и территории, зависимые от США, Бермуды
2	Европа (кроме России, Украины и Белоруссии), Средний Восток, Египет, Япония, Южная Африка, Свазиленд, Лесото, морские территории Франции, Гренландия
3	Юго-Восточная Азия, Южная Корея, Тайвань, Гонконг, Макао
4	Мексика, Центральная и Южная Америка, Карибский бассейн, Австралия, Новая Зеландия, Океания
5	Украина, Белоруссия, Россия, Африка (кроме Египта, Южной Африки, Свазиленда, Лесото и морских территорий Франции), Центральная и Южная Азия, Монголия, Северная Корея
6	Китайская Народная Республика, Гонконг
7	Зарезервировано для использования в будущем
8	Для международного использования [самолеты, круизные лайнеры и т.д. – *прим. пер.*]



› Через файловый браузер легко просмотреть файлы, открытые на общий доступ по локальной сети.

все мультимедиа в одном и том же приложении, и *Boxee* вполне по силам проиграть любимые фильмы с DVD после небольших изменений в ОС. Сначала нам нужно установить *libdvdread4*, библиотеку для чтения DVD:

```
~$ sudo apt-get install libdvdread4
```

Установив пакет, выполните следующую команду:

```
~$ sudo /usr/share/doc/libdvdread4/install-css.sh
```

Теперь нужно загрузить кодеки *w32codecs* – они предоставляют поддержку для множества форматов видеофайлов, в том числе DivX, Cinepak, H.263, MPEG 4, QuickTime, Real Audio/Video и Windows Media.

Вернемся в командную строку:

```
~$ sudo apt-get install w32codecs
```

Последний шаг – убедиться, что для DVD-проигрывателя корректно задан код региона. Вставьте свой любимый диск в привод и введите команду

```
~$ regionset
```

Если код региона, указанный после «drive plays discs from region(s):" [привод воспроизводит диски из региона(ов):], не соответствует вашему региону, его можно изменить. Чтобы сделать это, нажмите Y. Затем в ответ на предложение программы введите новый код региона и создайте новую маску.

После быстрой перезагрузки можно протестировать новую конфигурацию. По умолчанию, *Boxee* начнет проигрывать диск автоматически сразу после вставки в привод. Поэтому запустим *Boxee* и вставим диск с фильмом. Если фильм не начал воспроизводиться, предпримите несколько шагов по проверке настроек. Во-первых, у вас должна быть иконка DVD/CD. Если кликнуть по ней, воспроизведение должно начаться. Если так и произошло, следующий шаг – включить автоматическое воспроизведение. Зайдите в System [Система], затем Autoplay [Автоматическое воспроизведение] и убедитесь, что галочка Autoplay DVD-Video [Автоматическое воспроизведение DVD-видео] отмечена.

Общий доступ к файлам

Чтобы извлечь из имеющихся медиа все, нужно настроить общий доступ к файлам. Включив *Samba* (средство для общего доступа к файлам Windows) на своих компьютерах, вы сможете создать аудио- и видеопотоки в своей домашней сети. Так как протокол **smb** не зависит от платформы, им можно пользоваться в Windows, Mac или Linux.

Для включения общего доступа к файлам с помощью **smb** в Ubuntu потребуется установить *Samba*. В командной строке введите

```
~$ sudo apt-get install samba
```

После загрузки нужно будет создать себе пароль *Samba*. Пароли *Samba* хранятся не в **/etc/shadow** или **/etc/passwd**, а в **/etc/samba/smbpasswd**. Далее, **USERNAME** – ваше имя пользователя в Linux:

```
~$ sudo smbpasswd -a USERNAME
```

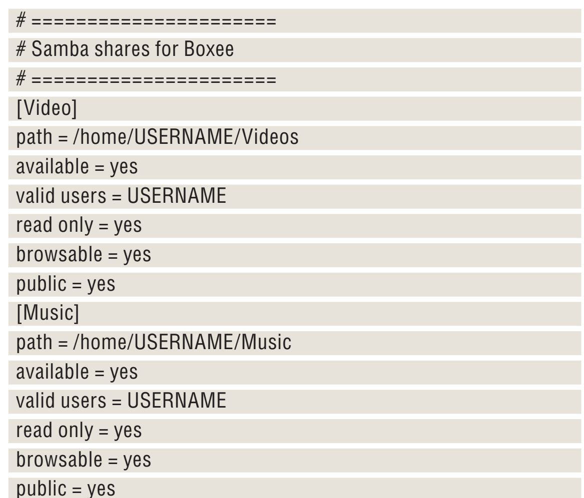
Задав пароль, приступим к созданию конфигурационного файла *Samba* для поддержки *Boxee*. Прежде чем менять **smb.conf** по умолчанию, неплохо создать его резервную копию.

```
~$ sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.bak
```

По умолчанию Ubuntu создает каталоги **~/Videos** и **~/Music**. Просто откройте общий доступ к ним. Чтобы это сделать, измените файл **smb.conf**:

```
~$ sudo gedit /etc/samba/smb.conf
```

В конец файла нужно добавить две новых секции. Скопируйте их из приведенного ниже фрагмента в свой **smb.conf**, затем замените **USERNAME** на свое имя пользователя.



```
# ======================
# Samba shares for Boxee
# ======================
[Video]
path = /home/USERNAME/Videos
available = yes
valid users = USERNAME
read only = yes
browsable = yes
public = yes
[Music]
path = /home/USERNAME/Music
available = yes
valid users = USERNAME
read only = yes
browsable = yes
public = yes
```

Сохраните обновленный файл и проверьте корректность настроек командой

```
~$ sudo testparam
```

После нажатия Enter вы должны увидеть список общих ресурсов, включая два новых. Чтобы изменения вступили в силу, придется перезапустить *Samba*. Проще всего сделать это командой

```
~$ sudo /etc/init.d/samba restart
```

Если у вас есть компьютеры с Microsoft Windows или Mac OS X, нужно открыть общий доступ к файлам и на них. *Boxee* может создать поток с любого компьютера локальной сети, на котором открыт общий доступ к файлам.

Universal Plug and Play (UPnP)

Если всем вашим контентом управляет сетевое хранилище данных (Network Attached Storage – NAS), вам повезло: *Boxee* поддерживает UPnP-устройства прямо из коробки. Большинство правильно настроенных устройств он обнаружит автоматически. Просто зайдите в Settings > Network > UPnP [Настройки > Сеть > UPnP]: устройство должно быть в списке. Если нет, добавьте его вручную, опять же зайдя в Settings > Media Sources > Manually Add Source [Настройки > Источники мультимедиа > Добавить источник вручную]. Источник добавляется через IP-адрес в следующем формате:

```
upnp://xxx. xxx.xxx.xxx
```

Затем выберите подходящие типы мультимедиа (Видео, Музыка или Фото), добавьте имя источника и укажите тип выполняемого сканирования. После этого источник должен появиться в разделе Files [Файлы] под именем, указанным при конфигурации.

По завершении настройки компьютеров-источников, создать поток – раз плюнуть. Чтобы сделать это из файловой системы, предоставленной в общий доступ, кликните в *Boxee* на Settings > Network [Настройки > Сеть]. В клиенте SMB введите имя пользователя и пароль общего ресурса, к которому хотите подключиться. Если компьютер находится в другой рабочей группе Windows, укажите ее имя. Вернитесь на основной экран и кликните на Files > Network [Файлы > Сеть]. Теперь должен появиться список доступных разделяемых ресурсов. Например, ТВ-программа, которую я смотрю, находится на **MNASSAL-ASUS/Video** (*Samba*).



› Интерфейс *Boxee* читается даже с расстояния в 10 футов, поэтому с диалогами настроек работать удобно.

По щелчку на имени ресурса будут показаны доступные папки и медиа-файлы. Кликните на одном из файлов – и вуаля: видеопоток запущен. Пока я это пишу, я смотрю одну из своих любимых ТВ-программ на маленьком HP Mini с Jolicloud. Видеопоток передается с ноутбука в соседней комнате с удивительно хорошей производительностью. Аудиопотоки создаются точно так же.

Через *Boxee* уже доступны подписки Netflix. Теперь *Boxee* расширяет эту идею. Этим летом *Boxee* запустит новую платежную платформу Payment Platform, которая предложит коммерческие программы по выбору и объединенные в пакеты. У контент-провайдеров появится возможность распространения программ в соответствии со спросом новых потребителей мультимедиа. Программы могут поставляться в пакетах, подписках или для одноразового платного просмотра. Ожидается, что цены будут ниже, чем в магазинах вроде iTunes – разница составит до 30 %.

В мире с постоянно растущим количеством норм и правил приятно видеть свежие решения, нацеленные на рост свободы потребителя. Хотя *Boxee* еще в развитии, это ведущее приложение в данном жанре. Подкасты, видео- и аудиопотоки в Интернете, воспроизведение локальных файлов, избранные пакеты/приложения, DVD, а вскоре – подписка и платный просмотр в приложении «все в одном», работающем в Linux, Windows и Mac. LXF

Коробка

Также на горизонте новый Boxee Box от D-Link. Box – это сетевое устройство, которое заменит небольшой настольный компьютер, Apple TV или ноутбук, подключенный к домашнему ТВ. Устройство подключается к локальной сети по Wi-Fi 802.11n или Ethernet, позволяя делиться контентом дома и через Интернет. В Box также есть Qwerty-клавиатура и пульт управления *Boxee*. Выход устройства запланирован на 2010 год и, если верить http://Mashable.com, его стоимость составит около $200.



DOSBox: Время

Тоскуете по классическим DOS-играм былых времен?
Адитья Шеваде покажет, как вернуться в юность, используя DOSBox.





Наш эксперт

Адитья Шеваде будущий инженер-электронщик и пользователь Linux с более чем четырехлетним стажем.

Помните ли вы того самого *Принца Персии*, *Worms* или *Lemmings*? С развитием компьютерных технологий эти классические игры потихоньку ушли, и, возможно, уже не запускаются на современных машинах.

Но если вас одолевает ностальгия, не грустите: решение есть. Такие игры были спроектированы для работы под старой ОС Microsoft Disk Operating Systen (MS-DOS), а ее можно воссоздать при помощи технологий виртуализации наподобие *VirtualBox*; но существует эмулятор, приспособленный для игр гораздо лучше.

DOSBox ориентирован на игры, хотя позволяет запускать и другое ПО. Он обеспечивает весьма точную эмуляцию звуковых карт, и вы можете настроить параметры видео, играть в многопользовательские игры и записывать скринкасты. Умно то, что он позволяет сбросить скорость эмуляции до уровня престарелой персоналки, максимально приблизив вас к древней DOS. Давайте выясним, как установить *DOSBox* и настроить его для наших нужд.

Двоичные сборки *DOSBox* для многих дистрибутивов Linux имеются на странице Downloads на сайте www.dosbox.com. Если нужной не нашлось, скомпилируйте ее сами из исходных текстов.

Параметры системы, где запускается *DOSBox*, будут влиять на скорость эмуляции. 400-Мгц Pentium II обеспечит производительность порядка процессора 386, а 1-ГГц Pentium III перенесет вас в высокопроизводительный 486-й процессор. Но, как и всегда, чем мощнее у вас система, тем лучший выйдет результат. Чтобы узнать об этом больше, посмотрите www.dosbox.com/wiki/System_Requirements.

Если вы решитесь собрать программу из исходных кодов, возьмите архив с сайта: http://prdownloads.sourceforge.net/dosbox/dosbox-0.73.tar.gz?download. Скачав файл, распакуйте его в отдельную директорию. Если у вас нет библиотеки *SDL*, установите ее, чтобы в *DOSBox* были звук и графика; загрузить ее можно с www.libsdl.org, но она наверняка найдется и в репозиториях вашего дистрибутива.

Затем установите *DOSBox*, используя следующие команды:

```
tar xvf dosbox-x.y.tar.gz
cd dosbox-x.y
./configure
make
make install
```

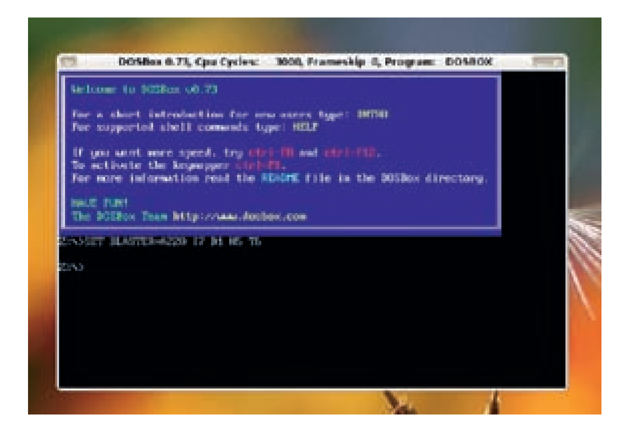
Если английский – не ваш родной язык, то на странице загрузок *DOSBox* доступно с десяток переводов на другие языки, включая русский.

Начнем

Успешно установив *DOSBox*, вы, вероятно, сразу же рванетесь поиграть. Ну и пожалуйста; но неплохо бы узнать об основном файле настройки, где можно подогнать *DOSBox* к вашим нуждам. Файл с именем **dosbox.conf** используется для глобальной настройки всех приложений, запускаемых вами в *DOSBox*. Если у вас есть локальный файл настройки, то он читается первым, а недостающие параметры берутся из основного. В Linux, конфигурационный файл создается при первом запуске программы в **~/.dosbox/**. Его имя обычно **dosbox-[номер_версии].conf**. Редактируя его, вы сможете настроить производительность *DOSBox* по своему желанию.

В этом файле несколько разделов. Более подробную информацию о них можно найти на вики: www.dosbox.com/wiki/Dosbox.conf. Вы можете указать, какой файл настройки загружать с каждым экземпляром *DOSBox*, передав аргумент в командной строке:

```
dosbox -conf путьКфайлуКонфигурации
```



› ***DOSBox*** **работает в окне на рабочем столе Linux и выдает информацию о производительности в своем заголовке.**

вспять



Попозже мы займемся изменениями, которые можно делать в файле настройки, но пока давайте запустим *DOSBox*.

Сперва вы увидите приглашение с буквой диска **Z:\>**. Последовательно нажимая Tab, вы получите список доступных команд. Отметьте для себя **mount**. Мы будем использовать ее для монтирования директории, содержащей ваши игры, и назначения ей буквы диска. Если все ваши игры находятся в **~/Games**, смонтируйте этот каталог в C:\, набрав:

```
Z:\>mount C /home/username/Games
```

Затем наберите **C:** для доступа к диску C, где теперь находятся все ваши игры. Или можно запустить *DOSBox* из командной строки Linux, передав в качестве аргумента имя директории – и она автоматически смонтируется как C:\.

```
dosbox ~/Games
```

Чтобы это происходило при каждом запуске *DOSBox*, добавьте вышеприведенную команду **mount** (и другие необходимые вам команды) в раздел **[autoexec]** файла настройки.

Монтирование оптических дисков требует похожих действий, но придется добавить несколько параметров, потому что CD-ROM'ы защищены от записи. Для настоящего CD допишите

```
Z:\>mount БукваДиска /media/МеткаОптическогоДиска/ -t cdrom
-usecd 0 -ioctl
```

А для монтирования образа –

```
Z:\>imagemount БукваДиска /Путь/К/Образу.iso -t iso.
```

Обратите внимание на особые аргументы, используемые для доступа *DOSBox* к оптическим носителям. Ключ **-t cdrom** активирует режим **MSCDEX**, позволяющий DOS-играм взаимодействовать с CD-ROM. Некоторые игры пытаются обратиться к CD, используя его метку, поэтому при монтировании вы можете передать **-label МеткаДиска**, если надо. Для лучшей производительности неплохо будет создать образ используемого CD и смонтировать его инструментом *imagemount*. Смонтировав CD, просто зайдите в директорию с игрой и играйте.

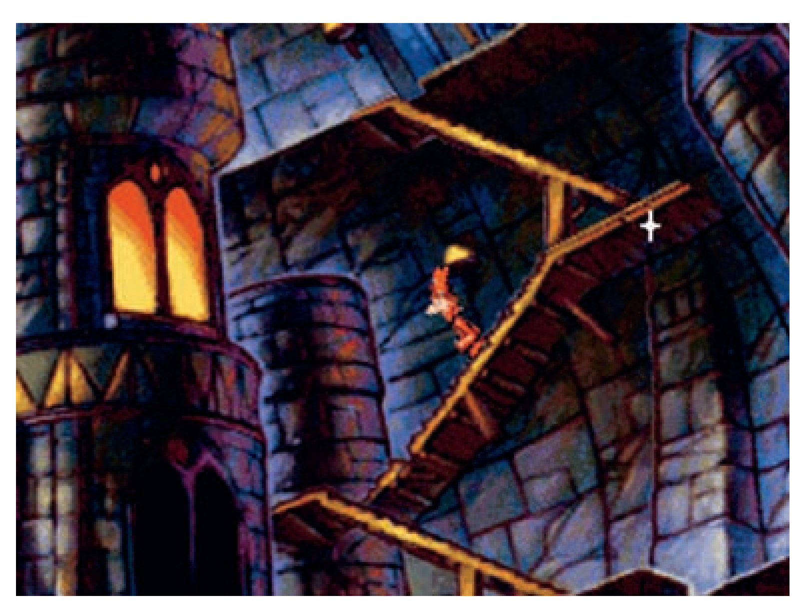
Чтобы достать виртуальный диск и освободить привод, вам придется его отмонтировать. В DOS нет команды umount – вместо нее наберите следующее:

```
mount -u БукваДиска
```

Ничего не слышу

DOSBox спроектирован для эмуляции нескольких звуковых устройств и позволяет DOS-программам взаимодействовать со звуковым оборудованием вашего компьютера для воспроизведения аудио. Так как во времена DOS не было унифицированных звуковых драйверов, настройка звука в *DOSBox* может оказаться непростой. Но здесь имеется поддержка большинства типичных устройств, включая динамик компьютера и различные версии SoundBlaster.

> «Помните ли вы того самого Принца Персии, Worms или Lemmings?»



› *Discworld* — одна из классических игр, которую можно откопать на ваших дисках и вернуть к жизни благодаря *DOSBox*. А также *SimCity* и *IndyCar Racing*.

Если звук отсутствует только в части игр, проверьте корректность установки последних. Если проблема не в этом, попробуйте выбрать в игре **soundblaster** или **soundblaster16** и убедитесь, что в файле настройки значится **address=220 irq=7 dma=1**. Вы можете также выбрать в качестве устройства для воспроизведения музыки MIDI на порту 330. Если звук все равно не слышен, проверьте свои колонки и убедитесь, что хост-система воспроизводит звук. Попробуйте также настроить значение циклов **core** на поменьше, в районе **3000**. Или попробуйте использовать менее популярные устройства, такие как SoundBlaster Pro (**sbtype=sbpro1** в файле конфигурации) или Gravis Ultrasound (**gus=true**).

Вы также можете столкнуться с трудностями вроде плавания частоты звука или его прерывания. В большинстве случаев это результат перегрузки процессора. Попробуйте уменьшить значение **core**, разрешить пропуск кадров или установить разрешение поменьше. Иногда помогает изменение частоты дискретизации звукового устройства. Кроме того, вы можете поэкспериментировать с параметром **prebuffer** в файле настройки и убедиться, что на хост-машине не запущено программ, пожирающих ресурсы или интенсивно обменивающихся с жестким диском. Подробная информация о настройке звука в *DOSBox* имеется на www.dosbox.com/wiki/Sound.

Скорость эмуляции *DOSBox* зависит от числа эмулируемых команд. Это называется «числом циклов под загрузкой» и может быть настроено через параметр **cycles**. Если вы планируете »



В процессе игры можно сделать снимок экрана, нажав Ctrl+F5. Он будет сохранен в **~/.dosbox/capture/**.

» **Пропустили номер?** Узнайте на с. 107, как получить его прямо сейчас.

запускать жадные до ресурсов игры (те, что вышли в последние годы жизни DOS), эмуляция их в реальном времени может перенапрячь *DOSBox*. В таком случае существуют различные пути для повышения производительности.

Первый и основной – закрыть все приложения, запущенные на вашей Linux-системе, чтобы освободить максимум ресурсов для *DOSBox*.

Улучшим производительность

По умолчанию значение **cycles** устанавливается автоматически. *DOSBox* использует обратную связь для определения, где игре необходимо больше циклов, и подстраивает производительность соответственно. Установка для **cycles** значения **max** в конфигурационном файле заставит *DOSBox* эмулировать игру с максимально возможной скоростью, независимо от необходимости. Однако установка на максимум может быть не оптимальным выбором, и поскольку настройки применяются на лету, лучше будет задать числа вручную. В оконном режиме вы увидите счетчик циклов в заголовке. Чтобы ускорить игру, нажмите Ctrl+F12, чтобы замедлить – Ctrl+F11. Ctrl+F9 увеличивает пропуск кадров, а Ctrl+F7 уменьшает.

Иногда бывает, что производительность улучшается, когда **cycles** установлены в **max**, а **core** – в **dynamic**. Однако это допускается только на архитектуре x86 и только в некоторых играх. Если игра, в которую вы играете, к ним не относится, вы можете увидеть существенное падение производительности.

Для улучшения производительности можно также поэкспериментировать с настройками звука – как в файле настройки, так и в самой игре. Попробуйте выключить звук или понизить его качество. Следующий код уменьшит частоту дискретизации, ускорив воспроизведение звука. **Prebuffer** задает число миллисекунд данных, сохраняемых сверх размера блока микшера.

```
[mixer]
rate=22050
oplrate=22050
rate=22050
pcrate=22050
tandyrate=22050
prebuffer=100
```

Дополнительный прирост производительности можно получить, изменив некоторые параметры в разделах **[sdl]** и **[render]** файла настройки. Вот как минимизируется постобработка видео, что заставит эмулятор работать быстрее:

```
[sdl]
fulldouble=false
fullresolution=original
windowresolution=original
output=surface
[render]
aspect=false
scaler=none
```

Скорая помощь

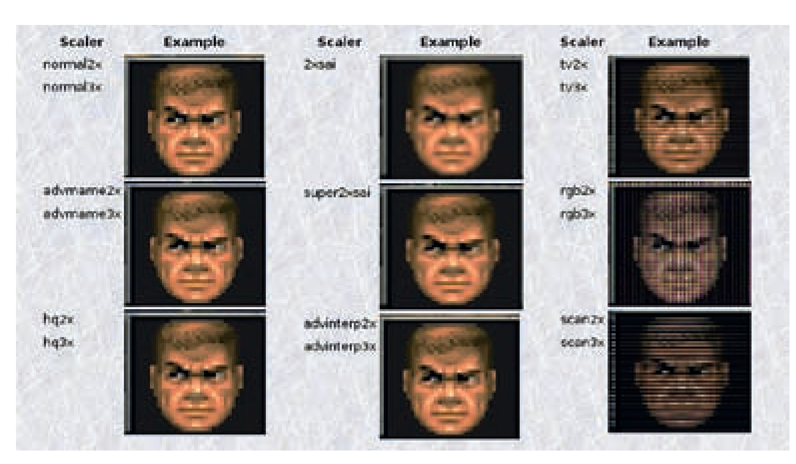
Если ваша мышь заблокирована в *DOSBox*, вы не можете использовать ее вне его. Для блокирования и разблокирования мыши нажмите Ctrl+F10 или щелкните где-нибудь в окне *DOSBox*.



› **Для записи скринкастов воспользуйтесь удобным настройщиком виртуальной клавиатуры в *DOSBox*.**

Плохое качество изображения обычно означает, что исходное разрешение слишком мало для вашего экрана, и обычно вы не можете радикально улучшить графику. Но можно поэкспериментировать с разделом **[render]** файла настройки. В нем есть параметр **scaler**. Старые игры в большинстве случаев спроектированы с учетом определенного (и по современным стандартам, низкого) разрешения. Основная причина этого была в недостатке вычислительной мощности. Когда вы растягиваете видео на ваш монитор с высоким разрешением, картинка выглядит ужасно. **Scaler** работает как фильтр, повышающий разрешение графики перед ее отображением, но это всегда делается после эмуляции. Поэтому все выполненные экранные снимки будут в низком разрешении, и вы сможете увеличить их позже, вручную, с помощью графического редактора. Детальный анализ доступных режимов масштабирования можно найти на www.dosbox.com/wiki/Scaler.

Наконец, если у вас мощный процессор, попробуйте включить режим двойной буферизации [**double buffering**]. В том же разделе параметр **full resolution** определяет разрешение в полноэкранном режиме. Для ЭЛТ-монитора попробуйте задать разрешение с множителем, кратным исходному разрешению игры, чтобы избежать проблем с частотой обновления. Или можно установить параметр **aspect** в **true**. Следующий параметр – это **output**. Режим **surface** по умолчанию отрисовывает графику в ее изначальном представлении. Если вы хотите получить чуть более современный вид, попробуйте применить **opengl** или **direct-draw**.



› **Помните того парня из *Doom*? Показаны различные параметры масштабирования, доступные в *DOSBox* для настройки производительности графики.**

Радость совместной игры

Одной из блестящих возможностей *DOSBox* является поддержка игры по сети: эмулируется клиент-серверная архитектура, где одна машина выступает как сервер, а все остальные подключаются к ней как клиенты. *DOSBox* может также эмулировать виртуальный модем, передавая весь трафик через IP. Это означает, что вам не потребуется дополнительного оборудования. Для начала загрузим IPX, добавив или изменив следующие строки в вашем файле настройки *DOSBox*.

```
[ipx]
ipx=true/false
```

Повторите этот шаг на всех компьютерах, принимающих участие в игре. Подключите их все и проверьте, что они видят друг друга из основной операционной системы. Затем на машине, которая будет сервером, наберите

```
ipxnet startserver [UDP-порт]
```

Параметр **UDP-порт** является необязательным и по умолчанию равен 213. При необходимости, вам следует открыть этот порт в хост-системе. Вы должны увидеть сообщение вида 'IPX Tunneling Server Started'. Обычно для Linux рекомендуется использовать порт с номером больше 1024: меньшие значения потребуют привилегий root и способны привести к конфликтам.

Вернитесь к клиентским машинам и скомандуйте

```
ipxnet connect <IP> [UDP-порт]
```

IP-адрес должен быть адресом сервера, а **UDP-порт** должен совпадать с таковым на сервере. Вам также нужно открыть этот порт на всех клиентских компьютерах. Кроме того, если ваши машины находятся за маршрутизатором, требуется изменить записи NAT.

Если все настроено верно, вы увидите сообщение 'IPX Tunneling Client Connected to Server at <IP>'. Для проверки состояния также могут быть использованы другие команды, такие как **ipxnet status** и **ipxnet ping**.

В итоге при входе в вашу игру вы должны увидеть все машины сети. Наигравшись, отключите клиенты, набрав **ipxnet disconnect** на каждом из них. Когда все клиенты отключатся, наберите **ipxnet stopserver** на сервере.

Скринкасты из DOSBox

Последняя версия *DOSBox* позволяет записывать все, что делается на вашем экране – например, если вам нужно потом показать кому-то, как завершить некий уровень игры.

Пользователи Linux нуждаются для записи видео в небольшой подстройке. По умолчанию сочетание клавиш для старта и остановки записи Ctrl+Alt+F5. Почти во всех случаях эти клавиши переключат вас в пятый виртуальный терминал [tty5].

Все, что вам нужно сделать – запустить *DOSBox Keymapper* с помощью

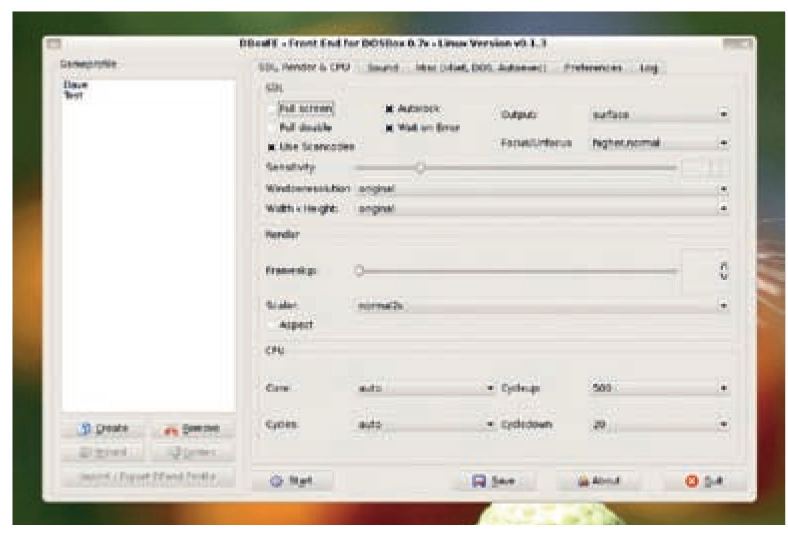
```
dosbox -startmapper
```

Вы увидите раскладку клавиатуры, показанную на предыдущей странице. Щелкните мышью на какой-нибудь клавише, чтобы увидеть физическую кнопку, соответствующую виртуальной. Щелчок на Video даст вам сочетание Mod1 + Mod2 как Hold и F5 как привязку. Проще говоря, вы должны нажимать F5, удерживая Mod1 и Mod2; и вам нужно изменить это сочетание. Давайте изменим его на Mod1 + Mod2 + V. Кликните на Video, затем на Add Button, и нажмите V. Это действие удалит Mod1 + Mod2 из Hold, поэтому щелкните на них снова, чтобы они удерживались.

Теперь вы можете записывать в *DOSBox* скринкасты. Разрешение и глубина цвета будут такими же, как и режим, эмулируемый *DOSBox*. Если вы измените настройки цвета или разрешение во время записи видео, текущий файл будет закрыт, и начат новый, с измененными настройками; он получит имя запущенного приложения. Скринкасты записываются в **~/.dosbox/captures**.

DOSBox – отличный инструмент, но его интерфейс командной строки может озадачить, особенно новичков. К счастью, есть немало графических интерфейсов, доступных на странице загрузок www.dosbox.com.

Нам очень нравится *DBoxFE*: он здорово упрощает редактирование конфигурационного файла. Вы можете также создавать для каждой игры отдельные файлы настройки, называемые «профилями игр». Последняя версия *DBoxFE* требует *Qt 4.3.x*. После успешной установки *DBoxFE* запустите его, и вы увидите окно, похожее на изображенное выше. Нажмите на Create [Создать], чтобы создать новый профиль, и дайте ему подходящее имя. Профили отображаются на левой панели. Во вкладках справа можно проделать все необходимые изменения. Они будут применяться при каждом запуске данного профиля.



› *DBoxFE* — один из графических интерфейсов для *DOSBox*, упрощающих его использование.

При внесении изменений важно указать расположение исполняемого файла *DOSBox*. Пройдите во вкладке Preferences [Настройки] к двоичному файлу *DOSBox* и выберите его. Его версия будет определена автоматически. Если вы плохо представляете, где находится данный двоичный файл, это можно узнать, набрав **which dosbox** в вашем терминале – вам вернется путь до исполняемого файла.

Далее, предположим, что после запуска данного профиля вы хотите монтировать и запускать *sampleGame*, расположенную в **/home/ИмяПользователя/Games/sampleGame**. Щелкните на вкладке Misc [Разное] и перейдите к Autoexec [Автозапуск]. Выберите директорию с игрой и букву диска, на который вы хотите ее смонтировать. Если игра идет на CD-ROM, можете указать и это. Закончив, нажмите на Add [Добавить] и после внесения всех изменений сохраните профиль и запустите *DOSBox* с ним, щелкнув на Start [Запуск].

> «При текущем наборе функций доступна почти вся классика.»

DOSBox – великолепный инструмент для всех, кто хочет вспомнить детство, но не готов собирать винтажный компьютер. Эмуляция делает процесс игры на новых системах более гладким и гораздо более быстрым. *DOSBox* пока не идеален, и ему не хватает ряда возможностей – например, сохранения состояния текущей сессии и ее последующего восстановления. Не помешал бы и собственный графический интерфейс, чтобы не полагаться на сторонних разработчиков.

Тем не менее, при текущем наборе функций без особых проблем доступна почти вся классика. При его активном и энергичном сообществе, *DOSBox* уже не остановить. LXF

Обновление папок

Если вы сделали изменения в директории, смонтированной в *DOSBox*, они не будут показаны автоматически. Образ придется обновить вручную. Для этого используйте событие Swap Image [Сменить образ], которое по умолчанию вызывается через Ctrl+F4.



Скорая помощь

Вы можете быстро переключаться между полноэкранным и оконным режимами, используя Alt + Enter (сочетание клавиш по умолчанию). В конфигурационном файле можно указать, что *DOSBox* должен стартовать сразу же в полноэкранном режиме.

R: Поиск законо

Что надо делать, чтобы быть хорошим преподавателем? Или: какими качествами надо обладать, чтобы занять видную должность в Ордене иезуитов? Ответы на оба вопроса дадут **Евгений Балдин**, **Владимир Лявшук** и **Сергей Петров**.

Студенты знают, что их ценят – следят за ними постоянно, и каждый экзамен становится праздником «раздачи слонов». С ними всё в порядке; ну, а кто оценит преподавателей? Говорят, анкетирование поможет. Проверим это!

Давным-давно в одной «далёкой-предалёкой галактике» провели анкетирование на тему «Преподаватель глазами студентов» и благополучно забыли об этом. После того, как с анкет была сдута пыль веков, выяснилось, что данные опроса представлены таблицами. Каждая строка опросной таблицы состоит из средней по группе студентов оценки по соответствующей графе анкеты. Строка также характеризуется курсом, предметом, преподавателем и кафедрой. Формат записей – CSV, то есть нечто вроде

```
Кафедра истории и философии;Галактическая история;Галактион Иванович Планетный; 5-й курс;4,6;4,7;4,3;4,8;4,6;3.6;2,7;...
```

Можно ли, оперируя этой информацией, поделить преподавателей на «хороших» и «плохих» – и, главное, как здесь применить *R*? Давайте попробуем разобраться.

Однажды в вузе

Для начала импортируем и объединим таблицы:

```
> data <- rbind(read.csv2("анкета1.csv"),
+           read.csv2("анкета2.csv"),
+            ...
+           read.csv2("анкетаN.csv"))
```

Увы, в реальной жизни никуда не деться от пропущенных значений, поэтому очистим от них данные опросов:

```
> data.clean <- na.exclude(data)
```

Воспользуемся методом основных компонент, чтобы уменьшить размерность анализируемых данных. График нагрузок основных компонент построим с помощью команды **prcomp**. В расчёте участвуют только ответы на вопросы анкеты:

```
> plot(prcomp((data.clean[,5:31]))
```

Как видно, здесь присутствует простая структура из двух факторов. Это довольно странно, учитывая, что в анкете было 26 вопросов. Иными словами, студенты давали 26 ответов, исходя лишь из двух предпосылок.

Давайте рассмотрим получившуюся простую структуру с целью идентификации двух выделенных в анализе факторов:

```
> biplot(prcomp(data.clean[,5:31]),
+     xlabs= rep( ".", length(data.clean[,3])))
```

Две взаимно независимые компоненты можно идентифицировать путем сопоставления преподавателей, участвующих в оценке, с местом, которое они заняли в пространстве факторов. Первая компонента отражает ось «Трудный предмет – простой предмет». Вторая компонента меняется в диапазоне «Добрый преподаватель – Строгий преподаватель». По соображениям этикета подробности этого процесса мы вынужденно опускаем.

У составителя анкеты, да и у оцениваемого преподавателя, естественно, возникает философский вопрос: «Что лучше: быть добрым или злым? И что делать, если тебе досталась сложная для студентов учебная программа?» Дело в том, что любое простое суммирование набранных преподавателями баллов с целью получить интегральную оценку будет эквивалентно проведению оси через построенное нами факторное пространство под определенным углом.

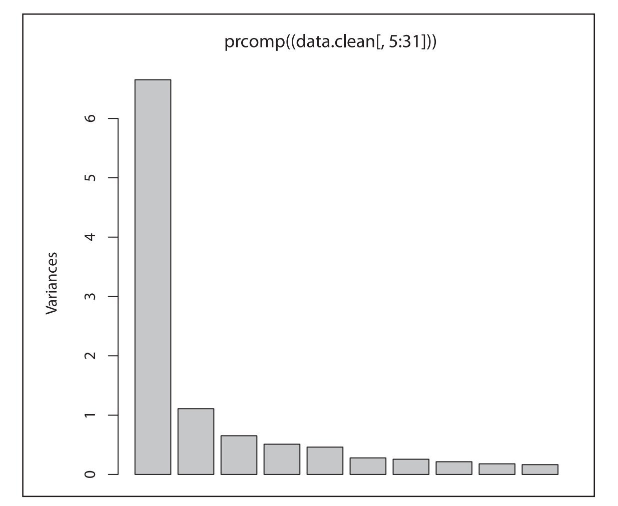


Рис. 1. График нагрузок основных компонент для задачи об оценке работы преподавателя.

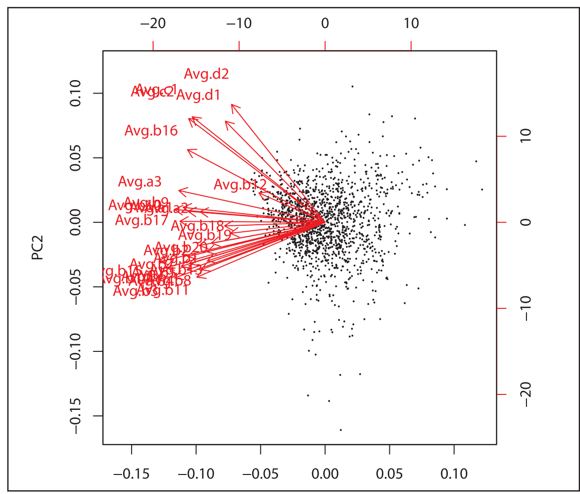


Рис. 2. Двойная диаграмма (biplot): вблизи начала координат те, кто ничем особенным не запомнился.

Месяц назад Мы ускоряли приложения R всеми доступными способами.

мерностей

На самом деле, мера, определяющая качество взаимодействия в системе «студент–преподаватель», имеется. Задумаемся над вопросом: «Кто из преподавателей получил оценки вблизи начала координат факторной плоскости?». Кто всегда получает среднюю оценку? Очевидный ответ: тот, о котором ничего особенного не помнят!

Если преподаватель взаимодействовал со студентом в процессе обучения достаточно сильно, то он оставил в его сознании чёткий след, способный отражать и доброту, и строгость, и сложность, и простоту... а также любое их сочетание! Иными словами, сила взаимодействия в системе «преподаватель–студент» – это длина вектора, проведенного из начала координат к точке на факторной плоскости, соответствующей анкете оцениваемого преподавателя.

Нам остается воспользоваться теоремой великого Пифагора

```
> result.pca <-prcomp((data.clean[,6:32]))$x[,1:2]
> vlijanie <- (result.pca[,1]^2 + result.pca[,2]^2)^0.5
```

Посмотрим, каково распределение полученной величины.

```
> plot(density(vlijanie)) #рисуем плотность распределения данных
> rug(jitter(vlijanie)) #добавляем по оси x штрихи значений данных
> boxplot(vlijanie,
+     add= TRUE,      #добавляем на тот же график
+     horizontal=TRUE, #располагаем boxplot по горизонтали
+     at= 0.2,        #позиция срединной линии boxplot
+     boxwex = 0.2)   #масштаб ширины boxplot
```

Оцененная сила воздействия преподавателя на студентов – неоднозначная величина. Безусловно, плохо, когда студенты «не помнят» преподавателя, но что делать с противоположным случаем? Может ли воздействие быть слишком сильным? Очевидно, по аналогии с другими рецепторами организма, любое чрезмерное воздействие приводит к повреждению и уж точно ощущается воспринимающим как дискомфорт.

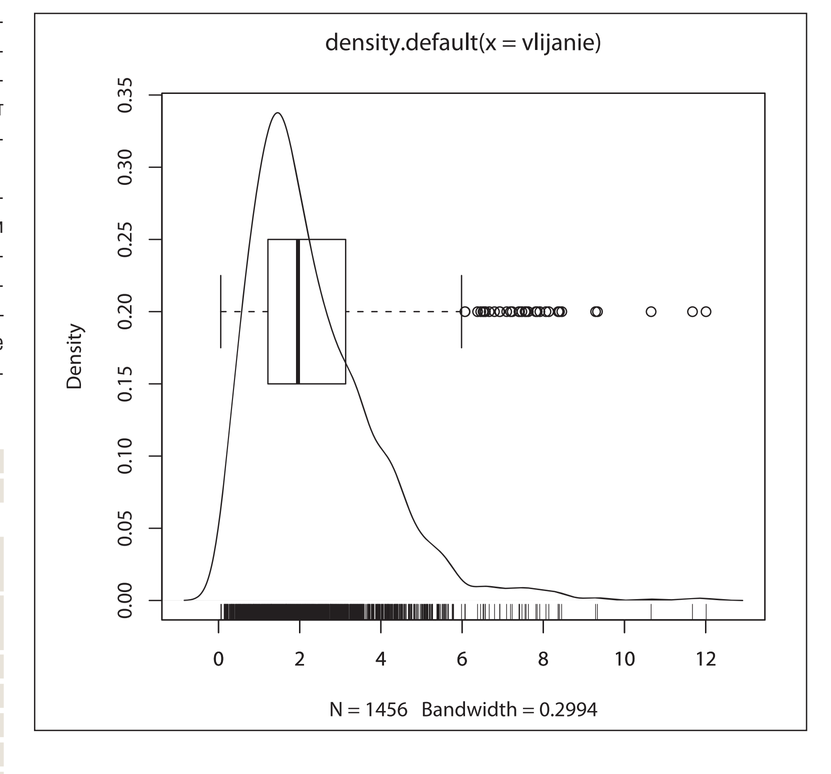
«Золотая середина» получается действительно в середине, и это порождает определённые трудности. Выделить «лучших», как обычно стремятся организаторы таких опросов, принципиально невозможно, так как они плавно перетекают в «не лучших».

Дело в том, что вопросы, задаваемые в анкете, не позволяют измерить то, о чём они напрямую спрашивают. Это крайне наивный, как показывает наш анализ, подход. Анкета позволяет лишь измерить некоторые объективные характеристики, исходя из которых отвечающий заполняет анкету. Анализируемый опросник позволит только узнать, кто из преподавателей вообще остался не замечен студентами. К ним, а также к тем, кто оставил в душе школяров слишком глубокий след, нужно присмотреться повнимательнее.

Орден иезуитов

Есть те, кто считает, что история – это не наука. Она, как правило, не говорит на языке математики и в какой-то мере даёт простор для спекуляций, то есть надуманных субъективных оценок. Объективность всячески приветствуется, но где её взять? Основную опасность для исторического исследования представляет искушение описать ушедшую реальность при помощи классификаций и понятий современности. Но такие вещи, как «государство», «экономика», «собственность» содержат внутри себя клише, сложившиеся в контексте современной культуры. Описание прошлого на основе применения к нему подобных клише чревато риском модернизации и ставит под вопрос валидность исторического исследования. Парадокс в том, что одни и те же данные, хранящиеся в архивных документах, допускают выдвигать и доказывать диаметрально противоположные гипотезы. В этом случае лучше послушать, что говорят сами данные, без нашей подсказки. »

«Одни и те же данные допускают противоположные гипотезы.»



› Рис. 3. Распределение «силы воздействия» преподавателя на студента: «добрый–злой», «легко–трудно».

	Pro Domo Vilnensi	Pro Domo Nesvisiensis	Pro Domo Varsaviensi
Zawistowski Franciscus	0	2	0
Dzieniszewski Albertus	0	0	0
Rymgayło Josephus	0	1	0
Kołakowski Martinus	0	1	2
Narmunth Nicolaus	4	0	2

› Рис. 4. Такую таблицу можно построить по архивным данным при помощи *awk* и *sed*.

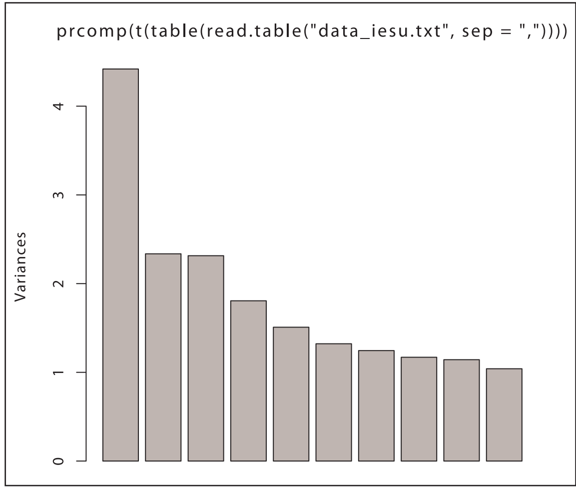
Не будем, как в случае анкет, сдувать архивную пыль – это вредно для здоровья. Возьмём в руки скучный документ. Его оригинал находится в Риме в Главном архиве Ордена иезуитов Archivum Romanum Societatis Iesu (ARSI). Пятый фонд в этом архиве называется Germania и касается жизни Германской Ассистенции Товарищества, куда входили также Польская и Литовская провинции ордена.

Том номер 130 из фонда Germania фактически представляет собой канцелярскую книгу, в которую для сведения генерала ордена заносились предложения провинциалов по кандидатурам на должность настоятеля определенного дома ордена и следующего провинциала. Таких записей с кандидатурами в период с 1684 по 1705 год зарегистрировано 412, то есть ведение книги начато при правлении генерала Шарля де Ноэля (1682–1686) и окончено со смертью генерала Тирсо Гонсалеса (1687–1705). Генералу, как правило, представлялась тройка кандидатур (по-латыни, terno), из которых он обычно выбирал первую по счету (отступлений от правил в книге зарегистрировано только 9). Если же ни одна из трёх предложенных кандидатур генерала не устраивала, то его канцелярия запрашивала у провинциала новое представление. Так произошло в 53 случаях из 412 (доля отвергнутых представлений равна 13 %), большинство из которых пришлось на конец правления де Ноэля и первую половину правления Гонсалеса, что свидетельствует об активном воздействии генералов на кадровую политику провинций.

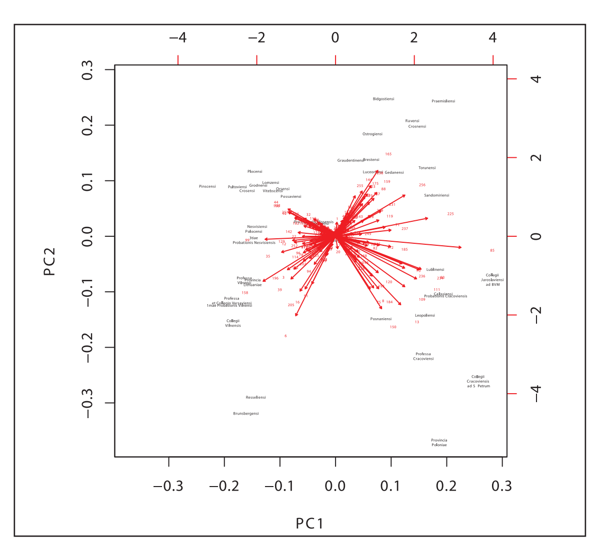
Данные в Germ.130 представлены в виде последовательных записей вида:

```
Pro rectoratu Vilnensi, 1684, Kitnowski Petrus, Krasnodebski
Adamus,Wyrwicz Andreas
```

Как видно, здесь указаны название должности, год и три кандидатуры, предложенные провинциалом. Иногда (крайне редко) провинциал предлагал менее трёх кандидатур за раз, и, видимо, отдельно обосновывал своё предложение. За двадцать лет так случилось только 8 раз (в двух случаях запись содержит две кандидатуры и в шести – только одну кандидатуру). Иногда, наоборот, провинциал предлагал больше, чем три кандидатуры на выбор. В период с 1684 по 1695 год так произошло 20 раз (16 раз по 4 кандидатуры, 3 раза по 5 кандидатур и 1 раз – 7 кандидатур). В последующие 10 лет ведения книги подобных случаев не зафиксировано. Примечательно, что несколько чаще нарушение правила выбора из трех кандидатур происходило в случае домов Польской провинции. Так, на должность провинциала Польской провинции предлагалось в 1687 г. 4 кандидата, в 1691 г. – 7 кандидатов, в 1695 – 5 кандидатов.



› Рис. 5. Руководители Ордена иезуитов при принятии кадровых решений руководствовались лишь тремя факторами.



› Рис. 6. Двойная диаграмма первых двух факторов в задаче о кадровых решениях.

Ищем зависимости

Используя утилиты обработки текстовой информации *awk* и *sed*, можно получить список пар «претендент–должность», причём должность фактически означает географическую привязку. Далее, мы свернули этот список в таблицу (небольшая часть которой представлена на рис. 4), по строкам которой расположились географически локализованные ректорские должности, а по столбцам – претенденты:

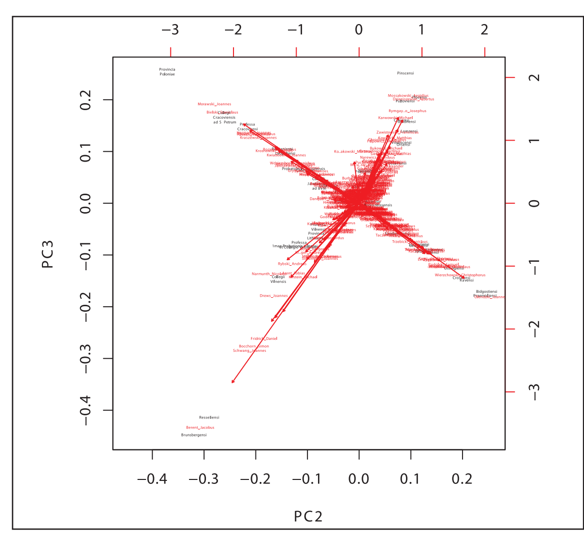
```
> library("xtable")
> iesu_table <- table(read.table("data_iesu.txt", sep=",")
+           )[c(137, 27, 112, 55, 92),2:4]
> colnames(iesu_table)<-c("Pro~Domo Vilnensi",
+                "Pro~Domo Nesvisiensis",
+                "Pro~Domo Varsaviensi")
> xtable(iesu_table,
>      align="lp{1.6cm}p{1.6cm}p{1.6cm}")
```

На пересечении строки и столбца находится количество выдвижений данного кандидата на данную должность за весь период наблюдений. Обработаем эту таблицу методом основных компонент.

```
> plot(prcomp(t(table(read.table("data_iesu.txt",
+                  sep=",")))))
```

Вскрытая анализом факторная структура состоит из трёх факторов. Давайте посмотрим на реальные данные в пространстве первых двух факторов (рис. 6).

```
> labelxs <- gsub("_",
+         " ",
+         rownames(t(table(read.table("data_iesu.txt",
+                   sep=",")))))
> labelxs <- sub("(Pro rectoratu) ", "", labelxs)
> labelxs <- sub("(Pro Domo) ", "", labelxs)
> labelxs <- sub("(tiae) ", "tiae\n ", labelxs)
> labelxs <- sub("(Pro regenda) ", "", labelxs)
> labelxs <- sub("(Provincia )", "Provincia\n", labelxs)
> labelxs <- sub("(Professa )", "Professa\n", labelxs)
> labelxs <- sub("(Collegii )", "Collegii\n", labelxs)
> labelxs <- sub("( ad )", "\n ad ", labelxs)
> biplot(prcomp(t(table(read.table("data_iesu.txt",
+                   sep=",")))),
+    cex=c(0.25,0.25),
+    arrow.len = 0.025,
```



› Рис. 7. Двойная диаграмма — но теперь для второго и третьего факторов.

```
+      ylabs = 1:length(t(table(read.table(“data_iesu.txt”,
+                              sep=“,”)))[1,]),
+      xlabs = labelxs,
+      )
```

Красным нанесены переменные, в которых оценивались должности. На основе их вклада и были выведены оси-факторы. Горизонтальная ось (первый фактор) отражает наличие естественной группировки – следовательно, это проявление ложного коэффициента корреляции. Данный корреляционный вклад вносят две несвязанные группы. Какие? По названиям коллегиумов, заглянув в энциклопедию (http://www.jezuici.krakow.pl/bibl/enc.htm), обнаружим, что они расположены в разных провинциях: польской и литовской. Обе провинции находятся на территории одного федеративного государства Речи Посполитой, в котором было несколько административных центров: Краков, Варшава, Вильно. Вертикальная ось отражает иерархию, существующую внутри провинций. Об этом свидетельствует концентрация руководящих должностей в столичных городах (Варшаве, Вильно и Кракове) в нижней части графика. Таким образом, выделенную ось мы можем с полным правом назвать «столичность–глубинка».

```
> biplot(prcomp(t(table(read.table(“data_iesu.txt”,
+                              sep=“,”)))) ,
+      choices= c(2,3),
+      xlabs = labelxs,
+      arrow.len = 0.025,
+      cex=0.25)
```

Третий фактор показывает стоящую отдельно группу должностей Пруссии, формально входившей в Литовскую провинцию. Это хороший повод отдельно рассмотреть факторную структуру Литовской провинции (подмножество из **data_iesu.txt**).

```
> labelxs <- gsub(“_”,
+               “ “,
+               rownames(t(table(read.table(“data_litwa.csv”,
                                  sep=“,”)))))
> labelxs <- sub(“(Pro rectoratu) “, “”, labelxs)
> labelxs <- sub(“(Pro Domo) “, “”, labelxs)
> labelxs <- sub(“(tiae) “, “tiae\n “, labelxs)
> labelxs <- sub(“(Pro regenda) “, “”, labelxs)
> labelxs <- sub(“(Provincia )”, “Provincia\n”, labelxs)
> labelxs <- sub(“(Professa )”, “Professa\n”, labelxs)
> labelxs <- sub(“(Collegii )”, “Collegii\n”, labelxs)
> labelxs <- sub(“( ad )”, “\n ad “, labelxs)
```

```
> biplot(prcomp(t(table(read.table(“data_litwa.csv”,
+                              sep=“,”)))),
+      arrow.len = 0.025,
+      xlabs = labelxs,
+      cex=0.25)
```

Мы видим слева Пинск, а справа – прусские Брунсберга и Решель. Что же отличает эти должности друг от друга? Национальная принадлежность кандидатов! Посмотрев на имена, вы сразу же заметите явную концентрацию славянских фамилий (оканчивающихся на «-ски») слева, а иностранных (немецко-звучащих) – справа.

Первый фактор явно отражает национальные различия – как в пространстве признаков, так и в пространстве значений. Полюса оси – это «прусскость» и «белорусскость».

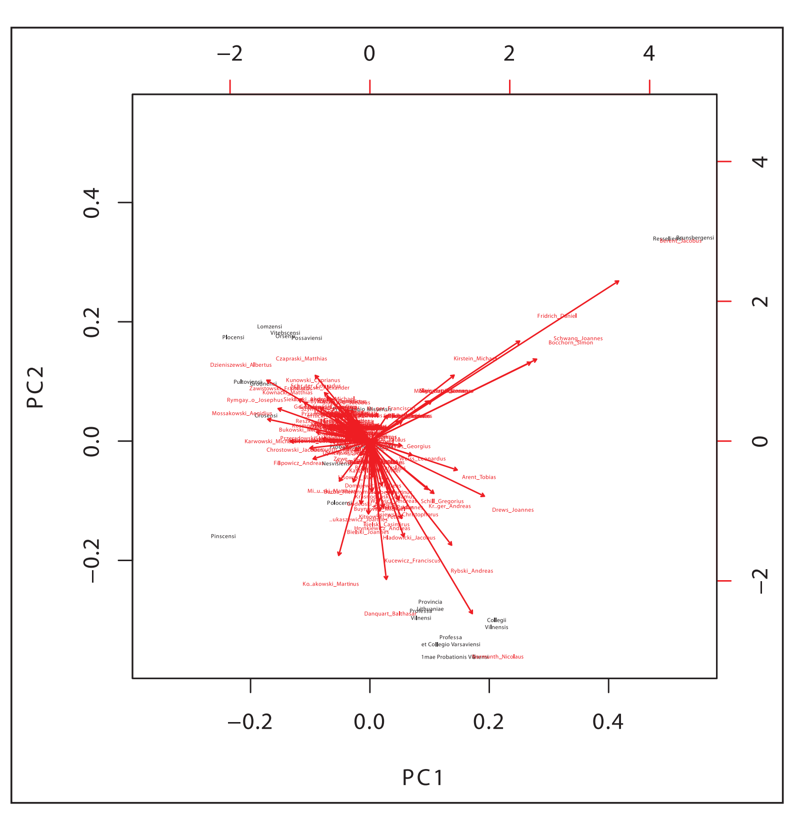
Второй фактор – знакомая нам «столичность–провинциальность», отражающая иерархию в организационной структуре провинции. Данный переход подтверждает спектр фамилий руководящих кадров ордена. Слева видны польские фамилии, а справа – не польские (чаще немецкие).

Эта же картина переносится на должности. В провинциальных городах немецкая часть спектра фамилий отсутствует вообще. В столицах же присутствуют и немцы, и поляки, причём они расположены согласно оси первого фактора.

И что с того?

Итак, что же вскрыла подобная обработка исторических сведений? Что представляет собой пространство факторов? Оно представляет собой пространство принятия кадровых решений (фрагмент соответствующей карты мира) в сознании провинциалов Ордена и двух генералов в период с 1684 по 1705 год.

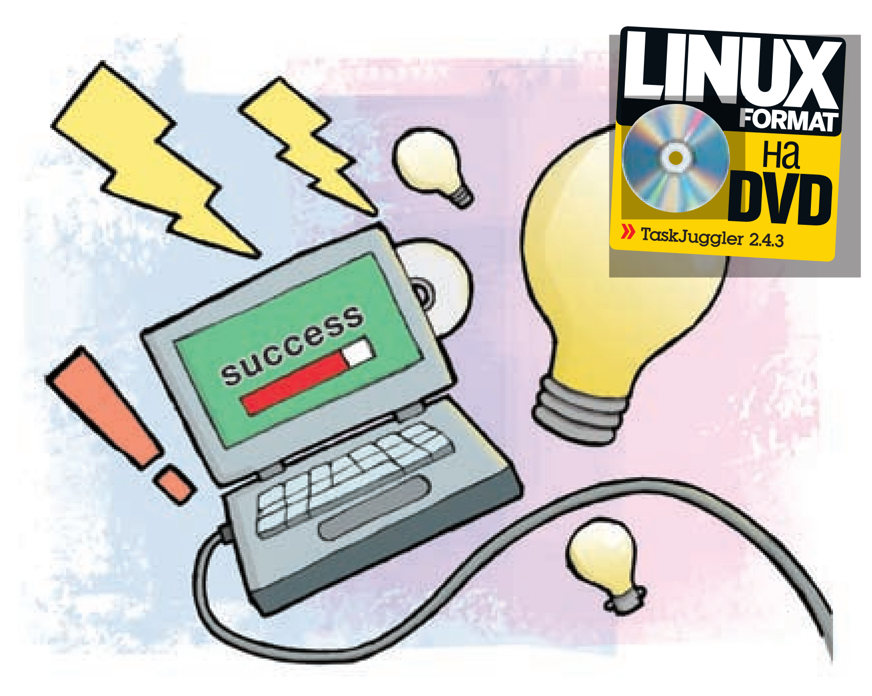
Что получили в результате историки? Фактическую информацию для проверки своих гипотез. Мы увидели зависимость, о которой никто никогда не говорил явно, но все учитывали, когда принимали решения. Естественно, её можно обнаружить и «методом пристального вглядывания». Статистика – всего лишь инструмент, но в ряде случаев – инструмент удобный. А R и Linux – подходящие инструменты для статистических исследований. LXF



› Рис. 8. Третий фактор, влиявший на принятие кадровых решений в Литве.

TaskJuggler: Рулим

Управление вещами гениальными и красивыми, проектами большими и маленькими упрощает *TaskJuggler*. **Шашанк Шарма** все разъяснит.





Наш эксперт

Шашанк Шарма пишет о свободном ПО уже более четырех лет – для разных изданий, включая Linux.com. Он один из соавторов Beginning Fedora от Apress.

Остановитесь на минутку и обратите внимание на движущихся вокруг вас людей. Одни куда-то собираются, другие откуда-то приезжают. И все это время они работают над проектом. Осознаете вы это или нет, но многое из того, что вы делаете каждый день, можно классифицировать именно так. И поскольку основную часть любого проекта составляют люди, вам не обойтись без эффективного средства управления проектами, которое помогало бы отслеживать людей и отдельные задачи, составляющие проект.

TaskJuggler – комплексное решение по управлению проектами, и вы можете использовать его от этапа планирования до завершения проекта. *TaskJuggler* умеет создавать всесторонние отчеты и упрощает управление задачами, затратами и ресурсами. Вы можете отслеживать достижения каждой из групп, работающих над проектом, и тут же определять, куда нужно направить побольше сотрудников, чтобы уложиться в плановые сроки.

Случайное свидание

TaskJuggler содержится в репозиториях большинства дистрибутивов, и вы, скорее всего, получите его командой

```
sudo apt-get install taskjuggler/
```

или

```
su -c yum install taskjuggler
```

Но тут есть подвох. Во-первых, *TaskJuggler* – это приложение KDE, а во-вторых, он разработан и испытан в первую очередь для OpenSUSE. Впрочем, особо не беспокойтесь: он работает безупречно и на других дистрибутивах.

Кроме того, версия 2.4.1, предлагаемая через репозитории для Jaunty, имеет большой недостаток. Она не работает, если вы находитесь в часовом поясе .30. Уязвимость касается *TaskJuggler* в целом, а не какой-то его конкретной функции, так что вы не можете использовать *TaskJuggler* 2.4.1 на .30 часовом поясе. Итак, если вы находитесь в часовом поясе +2,30 или +5,30, или аналогичном, лучше взять версию 2.4.2 или 2.4.3, последнюю.

TaskJuggler запускается из командной строки, но мы сфокусируемся на графической оболочке. Плюс, в терминале не видны графика и все красивые HTML-отчеты, требуемые каждому инструменту управления проектами, так что интерфейс вам в помощь.

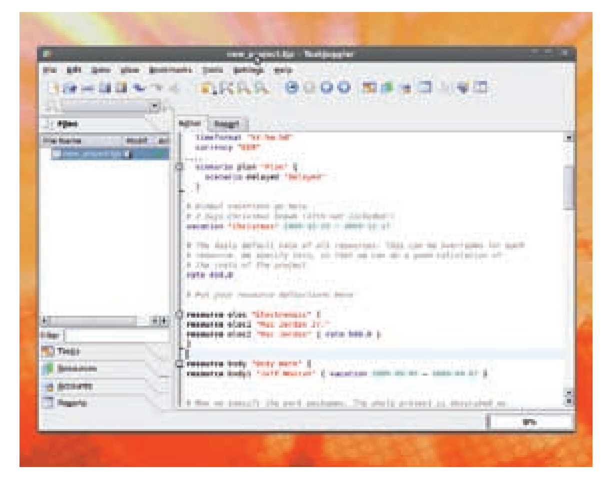
TaskJuggler предлагает множество встроенных ключевых слов (называемых свойствами – properties), которые можно применять для определения различных элементов вашего проекта. Например, свойством **currency** можно определить, будут ли расходы, описанные в проекте, указаны в долларах США, евро, или другой валюте, а свойством **project** — имя и ID вашего проекта; ну, и прочее.

Основная часть окна *TaskJuggler*, используемая для ввода всех сведений о проекте, называется редактором. На левой панели много объектов, таких как задачи (tasks), ресурсы (resources) и счета (accounts). Это удобно, если вам нужен быстрый доступ к конкретному разделу файла проекта.

Мы начнем эту статью с объяснения, как определить в *TaskJuggler* проект и все его тонкости, а затем покажем, как создавать графики, отчеты и т.д. и отслеживать выполнение вашего проекта.

Проект: начало

Чтобы начать новый проект, запустите *TaskJuggler* из меню Office [Офис] и выберите File > New [Файл > Создать]. Вам предоставят



› **Любое свойство *TaskJuggler* вместо полного имени можно обозначить через ID.**

» **См. также** *TaskJuggler* на реальном примере – в LXF125.

проектами

возможность начать пустой, простой или крупный проект. Если вы решите остановиться на пустом, придется вводить все вручную, а выбрав простой или большой, вы получите для своего проекта шаблон. Первым шагом является определение вашего проекта, подобное этому:

```
project carpimp "Pimping Cars" "1.0" 2009-08-20 2009-09-20 {
timeformat "%Y-%m-%d"
currency "EUR"
```

Это задает *TaskJuggler* имя вашего проекта, его ID, номер версии и даты начала и окончания. Свойства **currency** и **timeformat** говорят сами за себя. Вы можете использовать различные значения для различных проектов согласно вашим потребностям. Еще одно полезное свойство – **/scenario**. Обычно определяется два сценария: план пошаговой декомпозиции проекта и план компенсации задержек.

```
scenario plan "Plan" {
  scenario delayed "Delayed"
```

Здесь сценарий Delayed наследует все значения из сценария Plan, благодаря вложенной структуре. Это помогает отслеживать задержки в реализации проекта, если некоторые задачи не завершились за отведенное время. По мере развития нашего проекта все станет понятнее.

В качестве примера, на протяжении остальной части этой статьи, нашим учебным проектом будет тюнинг автомобиля. Да-с!

Определим ресурсы

Мы вернемся к задачам, которые составляют наш проект, через какое-то время, но сначала определим наши ресурсы. Все задачи используют их для достижения своих целей. Нашими ресурсами являются рабочие, потому что самим вам накладки на диски не смонтировать, если, конечно, вы не любитель крутить гайки.

```
resource elec "Electronics" {
resource elec1 "Mac Jordan Jr."
resource elec2 "Mac Jordan" { rate 500.0 }
}
resource body "Body Work" {
```

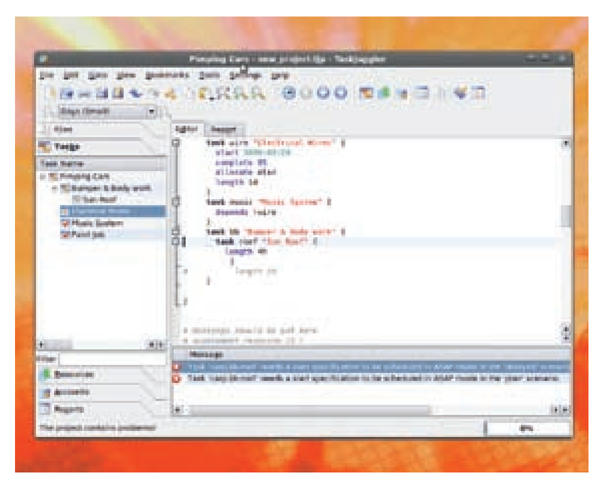
Использование макросов

Макросы используются в *TaskJuggler*, когда нужно повторить определенный текст много раз за проект. Например, если требуется распределить рабочих на разные задачи, вместо того, чтобы прописывать их на каждую, можно создать глобальный макрос и ссылаться в задачах на него.

В общем, если вы создадите макрос **EXPAND**, и затем сошлетесь на него через **${EXPAND}**, он раскроется в текст. Таким образом, при распределении ресурсов на разные задания можно создать следующий макрос:

```
macro allocate_elec [
allocate elec1 { rate 500.0 }
allocate elec2
]
```

и сделать на него ссылку в задаче **${allocate_elec}**.



› **При нажатии F9 для компиляции вашего проекта, *TaskJuggler* выдаст все ошибки внизу окна.**

```
resource body1 "Jeff Hoorah" {
vacation 2009-09-05 - 2009-09-07
workinghours {
wed 11:00 - 16:00
}
}
```

Мы только что определили две группы ресурсов, **Electronics** [Электроника] и **Body Work** [Кузовные работы]. При использовании любого из встроенных свойств необходимо указать для него имя и уникальный идентификатор. Для каждой команды мы определили рабочих и все их параметры. Например, свойство 'rate' [тариф] определяет расценки Мака Джордана, который поднаторел в электронике и, стало быть, вправе претендовать на более высокую заработную плату, чем другие люди, работающие над проектом. Каждый из ресурсов может иметь собственные исключения из различных определенных на глобальном уровне свойств. Скажем, в примере выше, **Body 1** (Джефф Ура) работает только в течение пяти часов, с 11:00–16:00, независимо от глобально определенного рабочего дня.

Скорая помощь

Обратитесь к руководству *TaskJuggler* и ознакомьтесь со всеми доступными свойствами.

Расходы и доходы

Немного математики, ребята. Ваш доход складывается из выплат вам, надеюсь, довольными вами клиентами, за вычетом расходов.

Употребление только что определенных нами ресурсов стоит денег. Добавим к ним стоимость всех примочек для оснастки автомобиля. Все это представляет собой расходы.

```
account eqp "Equipment" cost
account inc "Income" revenue
```

Свойство 'account' включает три значения: ID, имя и тип. Тип может быть тратой (cost) или доходом (revenue). Для анализа затрат, *TaskJuggler* вычитает сумму всех расходов из суммы всех доходов.

Таким образом, если у вас есть другой счет, «Pics», на который поступают доходы от продажи фотографий ваших тюнингованных автомобилей, прибыль будет рассчитываться как (Income + Pics) – (Equipment).

»

Итак, мы определили наши два счета. Главным шагом является назначение на них задач, с тем чтобы *TaskJuggler* отличал доходы от расходов. Мы остановимся на этом чуть ниже.

Ваш проект, несомненно, включает много задач. Смена покрышек, кузовные работы, покраска, установка супер-крутой музыкальной системы и т.д. – только некоторые задачи, входящие в тюнинг автомобиля.

```
task carp "Pimping Cars" {
  task pj "Paint Job" {
    depends !bb
  }
  task wire "Electrical Wires" {
    length 1d
  }
  task music "Music System" {
    depends !wire
  }
  task bb "Bumper & Body work" {
  task roof "Sun Roof" {
    }
  }
}
```

Для завершения этого проекта, мы должны предпринять несколько действий, каждое из которых описывается как задача. Некоторые из таких задач не могут начаться, пока не завершатся другие: например, **Paint Job** [Покраска] выполняется после кузовных работ. Здесь пригодится свойство **depends**.

Свойство **length** [продолжительность] определяет количество времени, отведенного на выполнение каждой задачи. Будучи определено в днях, **length** означает рабочие, а не календарные дни. То есть 1d означает 1 рабочий день, а не полные 24 часа. Вы можете определить продолжительность рабочего дня в вашем проекте как

```
dailyworkinghours 8
weekstartsmonday
  workinghours mon -fri 8:00 – 12:00, 13:00 – 17:00
  workinghours sat, sun off
```

Как видно из этого фрагмента, задачи могут быть вложенными. Это важно: не зная правильной структуры, сослаться на задачу нельзя. В нашем случае мы используем **!bb** как синоним для «Bumper & Body work» [«Бампер и кузовные работы»]. **!** означает переход к родительской задаче. Так, **!bb** при использовании в задаче «Paint Job» сперва поднимается к «Pimping Cars» (родительская задача – «Тачка на прокачку»), где мы находим «Bumper & Body work». Без **!** мы не можем сослаться на задачи вне текущей.

Для ясности представьте себе холодильник в кухне вашей квартиры. Если вы сидите в спальне и вдруг захотели выпить стакан сока, до холодильника вам не дотянуться. Вы сначала выйдете на кухню, а затем получите доступ к холодильнику.


Скорая помощь

Если вы не задали оплату на глобальном уровне, придется устанавливать ставки для каждого работника.

› Задачи не отображаются в календарном HTML-отчете, пока вы не определите для них ресурсы, используя свойство allocate.

› Диаграмма Ганта показывает процент выполнения работ по каждому проекту, задержки и сроки окончания.

Символ **.** аналогично используется для ссылки на подзадачи. Таким образом, чтобы вызвать задачу 'Sun Roof' [«Откидная крыша»], вам придется использовать **bb.roof**. Если вы когда-либо писали объектно-ориентированный код (скажем, на C++), смысл должен быть вам понятен.

Однако вы не сможете рассчитывать, что работа будет выполнена, если к ней не приставлен рабочий. Чтобы назначить работников, или ресурсы, как их называет *TaskJugger*, воспользуемся свойством **allocate**.

```
task wire "Electrical Wires" {
  allocate elec
  length 1d
}
```

Разделенный запятыми список работников может быть использован со свойством **allocate** для назначения на задачу нескольких ресурсов.

Тщательность в деталях

TaskJuggler ожидает, что прежде чем компилировать проект, вы введете всевозможные детали для каждой из ранее определенных задач.

Одно из наиболее важных свойств для каждой задачи – **start**, оно используется для указания даты *TaskJuggler*, на которую задача должна быть начата. Но, несмотря на самые продуманные планы, некоторые вещи начинаются с опозданием. Например, если задача Music System фактически начинается 2 сентября, а не 1 сентября, вы можете сообщить об этом *TaskJuggler* вот так:

```
task music "Music System" {
  depends !wire
  start 2009-09-01
  delayed:start 2009-09-02
  length 2d
}
```

Помните, как мы в начале создали сценарии **plan** и **delayed**? Мы здесь используем **delayed**, чтобы указать фактическую дату начала задачи. Можно также использовать **delayed** для других припозднившихся задач.

По умолчанию, *TaskJuggler* считает все задачи идущими по графику и выполняемыми в срок. Если это не так, можно указать *TaskJuggler* статус каждой задачи.

```
task music "Music System" {
  depends !wire
  start 2009-09-01
  delayed:start 2009-09-02
  length 2d
  delayed:length 5d
  complete 45
```

Свойство **complete** сообщает *TaskJuggler*, что задача Music System завершена только на 45 %. Также обратите внимание, что мы использовали выше два свойства **delayed**, для определения задержки с началом и продолжительностью.

Помимо простого выделения ресурсов для задач, можно также указать количество часов, в течение которых они будут затребованы. Мы используем свойство **limits**:

```
task wire "Electrical Wires" {
allocate elec2 { limits { dailymax 3h } }
length 4d
}
```

В этом фрагменте мы сообщили *TaskJuggler*, что наш ресурс **elect2** (Мак Джордан) за каждый из четырех дней выполнения своей задачи будет работать с электропроводкой только по три часа.

Покажи мне деньги

Последнее, что нужно сделать – это назначить свои задачи на надлежащие счета. Так как мы создали счет **eqp**, куда списываются затраты на оборудование, используемое под каждую задачу, давайте использовать его глобально, чтобы все задачи могли к нему обращаться:

```
task carp "Pimping Cars" {
account eqp
task pj "Paint Job" {
}
task wire "Electrical Wires" {
}
task music "Music System" {
}
task bb "Bumper & Body work" {
}
}
```

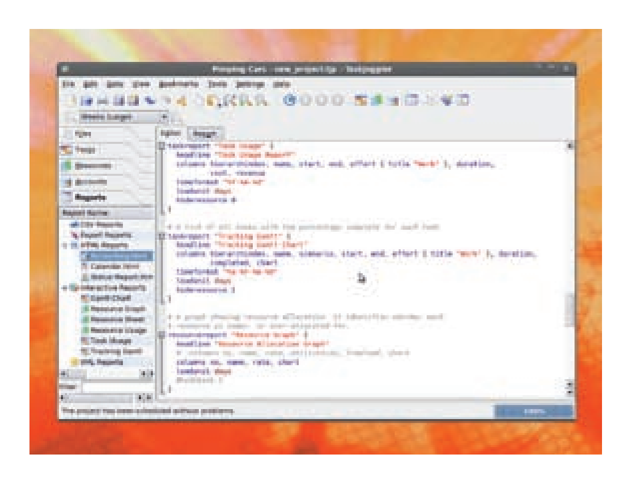
Далее, мы должны показать все наши доходы. Чаще всего по проекту выдается начальный аванс, потом на различных этапах развития продукта принимаются платежи, а уж потом, по завершении, выполняется окончательный расчет.

Разумеется, вам придется вписывать все эти платежи. Для этого надо определить в *TaskJuggler* отдельные этапы, на которых демонстрируется продукт:

```
task demo "Demonstrations" {
  account inc
  task down "Downpayment" {
    # Представлены детальный план тюнинга и
раскладка по времени
    credit 2009-08-25 "Initial downpayment" 5000.0
  }
  task fd "First Demonstration {
    # Заказчик оценивает результаты кузовных работ
    depends !!carp.bb
    startcredit 15000.0
  }
task finish "Finished" {
   # Проект завершен, тачка заказчика
прокачана по самое не балуй
    depends carp.pj
    startcredit 20000.0
  }
}
```

Вместо указания кредита по ходу проекта наряду с другими выплатами, вы можете сделать это в самом начале, где впервые определяются разные счета:

```
account eqp "Equipment" cost
account inc "Income" revenue {
  credit 2009-08-25 "Initial downpayment" 5000.0
}
```



› **Явное преимущество выбора шаблона при старте новых проектов теперь стало очевидным, а?**

Если цифры кажутся вам завышенными, мы тут ни при чем. На тюнинге автомобилей и правда делаются большие деньги. Но если вы не уверены, начните с малого. Скажем, с тюнинга мотоциклов.

Есть три свойства, которые можно использовать при определении сумм, заплаченных вам вашими клиентами: **credit**, **startcredit** и **endcredit**. **Startcredit** и **endcredit**, как следует из названия, используются для определения выплат, сделанных вам в начале и по окончании задачи. Свойство **credit** используется для задания первоначальных выплат вам.

Отчетность

К счастью, после титанических усилий для подгонки *TaskJuggler* ко всем сложностям вашего проекта, для создания отчета ничего специального делать не придется.

Если при старте проекта вы выбрали простой или большой шаблон, у вас уже есть все, чтобы просмотреть отчеты по вашему проекту. Просто нажмите F9 для компиляции проекта и выберите вкладку Reports [Отчеты] на левой панели. Теперь нажмите на любом варианте из перечисленных – и вуаля, вот ваш отчет.

Поскольку без диаграммы Ганта ни один отчет не полон, можно создать и ее, с помощью такого кода:

```
taskreport "Gantt Chart" {
  headline "Project Gantt Chart"
  columns hierarchindex, name, start, end, effort, duration, chart
# В этом отчете мы хотим видеть перед датой
сокращенное название дня недели. За это отвечает %a.
  timeformat "%a %Y-%m-%d"
  loadunit days
  hideresource 1
}
```

Теперь, скомпилировав проект, вы увидите ссылку Gantt Chart [Диаграмма Ганта] в интерактивных отчетах в меню Отчеты. Щелчок по ней вызовет отчет с указанием всех заданных нами полей. Диаграмма Ганта отображается на панели в правой части окна *TaskJuggler*.

Помимо **taskreport**, можно также заставить *TaskJuggler* создать отчет по использованию ресурсов, прибегнув к помощи **resourcereport**. Аналогично можно применить **htmlaccountreport** для контроля счетов:

```
htmlaccountreport "Accounts.html" {
columns no, name, scenario, total, monthly
headline "Income and Expenses"
caption "Income and Expenses at a glance. But, also in
detail at the same time. You'll see from the screenshot."
scenarios plan, delayed
}
```

Строка **scenarios plan**, **delayedstatement** поможет быстро заметить разницу в расходах в случае задержки. **LXF**

Python: Рисуем

Clutter предоставляет интерфейс, но, опираясь на свой опыт в графике, **Ник Вейч** добавит к нему мощь *Cairo* и создаст сетевые часы.





Наш эксперт

Ник Вейч писал Python-приложения еще до рождения *Linux Format*, и все это было лишь преддверием к настоящему.

Мы уже видели, как создавать в *Clutter* простые объекты, размещать их на экране и даже анимировать. Но пока мы создавали при помощи *Clutter* одни прямоугольнички. Эй! А как же другие формы и поверхности? Не-е, только не в *Clutter*. Но голь на выдумки хитра: *Clutter* запряг вместо себя библиотеки *Cairo*. Они содержат дюжину операций рисования в текстурах, и мы можем создать контекст *Cairo* и воспользоваться ими. Когда *Cairo* выполнит сложную часть, *Clutter* применит результат, как любую другую текстуру.

В ранних версиях *Clutter* поддержка *Cairo* была реализована в виде встраиваемого модуля. Теперь это часть основного кода *Clutter*, и нам не придется импортировать какие-либо специальные библиотеки – хотя для работы с контекстами *Cairo* потребуется Python-модуль *Cairo*. Откройте интерактивную оболочку Python, введя **python** в командной строке, и наберите следующее:

```
>>> import clutter
>>> import cairo
>>> import math
```

Если вы планируете работать с окружностями, дугами и так далее, имеет смысл импортировать модуль **Math из стандартного комплекта Python:** в нем имеется константа пи (**math.pi**) и методы для преобразования радиан в градусы и обратно. Далее выполним стандартный ритуал настройки сцены *Clutter:*

```
>>> stage = clutter.Stage()
>>> stage.set_size(500,500)
>>> blue =clutter.Color(0,0,255,255)
>>> stage.set_color(blue)
```

Если вы следили за серией уроков по *Clutter,* то понимаете, что это обычные действия, выполняемые при подготовке сцены. Теперь создадим нашего актера.

```
>>> cairoactor=clutter.CairoTexture(50,50)
>>> cairoactor.set_position(225,225)
```

Здесь генерируется специальный актер, получающий текстуру из *Cairo*, и мы размещаем его слева от центра нашей сцены, с учетом размера круга. А теперь нарисуем объект.

```
>>> context=cairoactor.cairo_create()
>>> context.set_operator(cairo.OPERATOR_OVER)
>>> context.set_source_rgb(1,1,0)
>>> context.arc(25,25,20,0, 2*math.pi)
>>> context.close_path()
>>> context.fill()
```

Объект **context**, созданный нами здесь, на самом деле является измененным объектом *Cairo* и наследует все полезные методы *Cairo*. Прежде чем начинать любые операции рисования, надо настроить режим рисования. Чаще всего мы будем использовать операцию **OVER**, которая рисует поверх всего, что уже имеется в текстуре. **Set_source_rgb()** устанавливает цвет пера для будущих операций. Затем изобразим наш круг.

Круг, в терминах *Cairo*, это просто дуга, простирающаяся на 360° (или 2π, поскольку *Cairo* измеряет углы в радианах). Для этого метода мы должны указать координаты X и Y центральной точки, затем радиус дуги, а также начальный и конечный углы в радианах. Мы указали от **0** до **2*math.pi**, то есть 360°.

Заливаемый контур должны быть замкнутым. Вы можете подумать, что он уже замкнут – а вот и нет. По мнению *Cairo*, это пока

Что такое Cairo?

Не считая родного города пирамид, *Cairo* – это название графического инструментария, созданного Карлом Вортом [Carl Worth] и другими. Мы брали у его разработчиков интервью в **LXF111**.

Целью *Cairo* было создание стандартного набора графических инструментов для выполнения насущных задач быстро и просто. Во многих приложениях, где надо отрисовывать графику, реализуются свои собственные алгоритмы – в уверенности, что они будут быстрее, лучше и проще в сопровождении; но часто это не так.

Cairo кросс-платформенный, интегрируется с несколькими графическими оболочками и поддерживает графическое ускорение. Дополнительную информацию см. на www.cairographics.org.

» **Месяц назад** Мы создали видеопоток с помощью *GStreamer* и *Clutter.*

с Clutter



просто контур, который начинается и заканчивается в одной точке. Чтобы замкнуть его, следует вызвать метод **close_path()**, а уж потом можно закрашивать. Метод **fill()** пытается заполнить текущую активную фигуру. Если же мы хотим изобразить контур фигуры (то есть окружность, а не круг), следует использовать метод **stroke()**.

Создав круг, отобразим этот объект, добавим его в сцену и затем отрисуем все вместе:

```
>>> cairoactor.show()
>>> stage.add(cairoactor)
>>> stage.show_all()
```

Ну и где круг?! Мы же ввели команды рисования, а затем добавили актера на сцену! Ответ в том, что мы покамест в контексте *Cairo*. Модуль *Cairo* не приступит к выводу изображения вашей текстуры, пока вы его не удалите. Да, это звучит странно, но попробуйте:

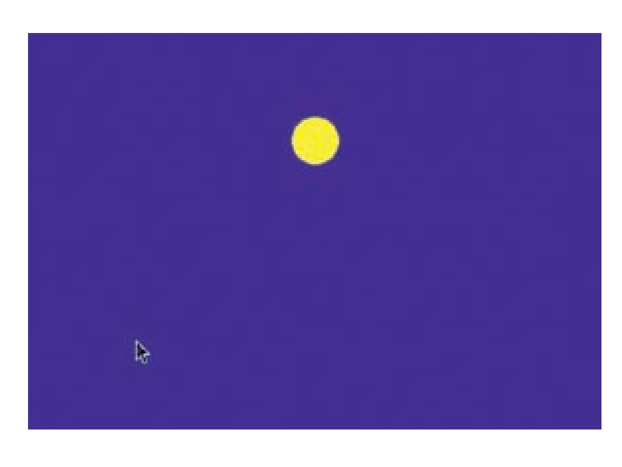
```
>>> del context
```

Как это работает

Как по волшебству, круг появился! Запутаться немудрено, но представьте себе такую модель. Пусть *Clutter* – директор картинной галереи. Галерея – то место, где объекты отображаются; но создаются они в другом месте. Галерея (*Clutter*) берет холст (текстуру) и отправляет его художнику (*Cairo*). Затем галерея просит художника нарисовать картину в соответствии с инструкциями (вызовы в контексте). Но чтобы галерея могла отобразить ее, художник должен отправить картину обратно (удалить из контекста). Если галерея когда-либо захочет изменить картину, она может просто вернуть ее художнику (создать для объекта новый контекст). Так что воспринимайте удаление объекта контекста как завершение контракта.

Есть и другие моменты, о которых следует знать при работе с кодом *Clutter/Cairo*, особенно в Python. Текстура обновляется, когда контекст выходит за границы области видимости (в коде). В предыдущем примере это происходило потому, что мы явным образом удаляли его, а Python очень хорошо очищает память, которая больше не требуется. Также, контекст может выйти из области видимости и автоматически удалиться, если, например, это локальная переменная в завершившейся функции. За этим необходимо следить. Некоторые программисты используют эту особенность, чтобы прибирать за собой – ведь если сборщик мусора Python желает сам заботиться об удалении ваших объектов, то чего напрягаться? Но иногда наличие явных удалений облегчает понимание кода. К тому же для некоторых библиотек, несмотря на то, что Python освобождает память, модуль может не освободить дескриптор контекста, что вызовет проблемы.

Большинство операторов рисования *Cairo* имеют альтернативы, использующие относительные координаты вместо абсолютных. И это замечательно, когда объект рисуется и точно не известно, где он начинается, или известно лишь, что необходимо перейти в определенном направлении на заданное расстояние (к сожалению, сложение векторов среди умений *Cairo* не значится).



› Ух ты! Поискав, вы найдете немало любопытных web-камер. Или возьмите свои файлы.

Чтобы разобраться со всем этим, изобразим треугольник. Помните, что координаты, передаваемые методам **context**, задаются относительно текстуры *Cairo*, а не сцены.

```
>>> import clutter
>>> import cairo
>>> stage = clutter.Stage()
>>> stage.set_size(500,500)
>>> blue =clutter.Color(0,0,255,255)
>>> stage.set_color(blue)
>>> cairoactor=clutter.CairoTexture(50,50)
>>> cairoactor.set_position(225,225)
>>> context=cairoactor.cairo_create()
>>> context.set_operator(cairo.OPERATOR_OVER)
>>> context.set_line_width(5)
>>> context.set_source_rgb(1,0,0)
>>> context.move_to(3,3)
>>> context.line_to(47,3)
>>> context.rel_line_to(-22,44)
>>> context.close_path()
>>> context.stroke_preserve()
>>> context.set_source_rgb(0,1,0)
>>> context.fill()
>>> del context
>>> cairoactor.show()
>>> stage.add(cairoactor)
>>> stage.show_all()
```

Поскольку на сей раз мы применили операцию **stroke**, необходимо установить толщину линии. Кроме того, мы использовали метод **stroke_preserve()**. Опцию **preserve** имеют большинство методов рисования. Обычно их вызов удаляет активный контур в ходе стандартной процедуры очистки текстуры. Если требуется использовать контур вновь, сохраните его для следующей операции опцией **preserve**. Если запутались с координатами, смотрите рисунок на следующей странице.

»

Рисуем часы

Для наших часов нарисуем циферблат при помощи объектов *Cairo*, затем создадим несколько текстур для стрелок. Стрелки можно сделать любой сложности. В нашем проекте мы применим вытянутые ромбы, поскольку сложные стрелки требуют больше кода для рисования и отнимают место на объяснение происходящего.

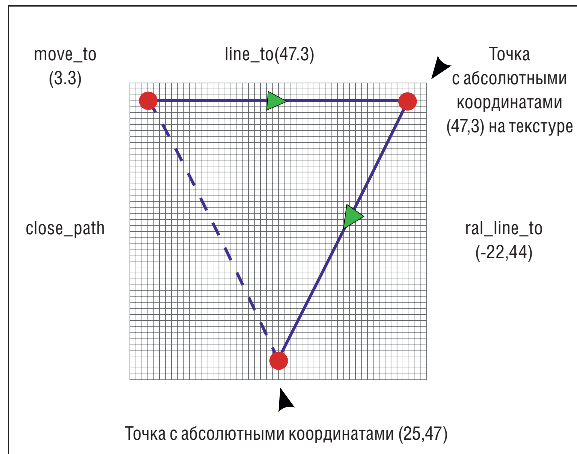
Циферблат часов – это текстура, но в ней будут повторяющиеся по кругу объекты, обозначающие точки циферблата. Можно было заняться математикой и в цикле разместить каждую цифру в виде отдельной текстуры. Но это утомительно и занимает много места; вместо этого создадим их программно.

Облегчим себе жизнь некоторыми преобразованиями из *Cairo*. Метод **translate()** смещает центральную ось контекста области рисования, а **rotate()** вращает ее. Скомбинировав их, мы можем повторно рисовать один и тот же круг и сдвигать контекст между отрисовками – воспринимайте это как печать на листе бумаги, который сдвигается перед очередным оттиском. Штамп опускается на то же место, а поверхность смещается. Мы разбили код на разделы, чтобы вы могли видеть это в действии, начиная с **def_drawface**.

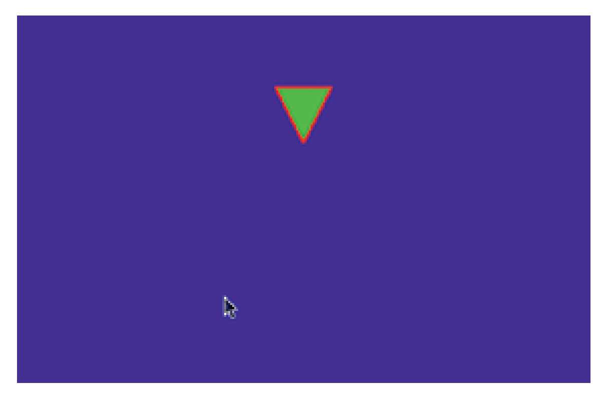
Запускаем часы

Данный код создает объект класса **clock**, содержащий все необходимые причиндалы. Процедура **init** настраивает сцену, вызывает функции для отрисовки необходимых текстур и реализует шкалу времени. Это та шкала времени, что заставляет наши часы тикать – она обновляет их каждые 1000 мс. Отметим, что метод **update** обращается к системному времени, не полагаясь только на прерывания, так что часы всегда синхронизированы с системными.

```
import clutter
import cairo
import math
import time
class clock:
  def __init__ (self):
    self.stage = clutter.Stage()
    self.stage.set_color(clutter.Color(0, 0, 0, 255))
    self.stage.set_size(500, 500)
    self.stage.set_title('LXF Atomic Clock')
    self.stage.connect('key-press-event', self.parseKeyPress)
    self.stage.connect("destroy", clutter.main_quit)
    self.drawface()
    self.drawhands()
    self.stage.show_all()
    self.t=clutter.Timeline()
    self.t.set_duration(1000)
    self.t.set_loop(True)
    self.t.connect('completed', self.update)
```



› **Вызовы контекста и их действие. Синим цветом показан контур.**



› **Сохранив контур при помощи метода preserve, можно выполнять в Clutter многократные операции рисования.**

```
    self.t.start()
    clutter.main()
  def parseKeyPress(self, stage, event):
    # Выполняем действия по нажатию клавиши пользователем
    if event.keyval == clutter.keysyms.q:
      # Если нажата клавиша "q" - выходим из приложения
      clutter.main_quit()
  def drawface(self):
    self.face=clutter.CairoTexture(500,500)
    self.face.set_position(0,0)
    context=self.face.cairo_create()
    context.set_operator(cairo.OPERATOR_OVER)
    context.set_source_rgb(1,1,0)
    context.translate(250, 250)
    for a in range(12):
      context.arc(0,230,12,0, 2*math.pi)
      context.close_path()
      context.fill()
      context.rotate(2*math.pi/12)
    for a in range(4):
      context.arc(0,230,18,0, 2*math.pi)
      context.close_path()
      context.fill()
      context.rotate(2*math.pi/4)
    del context
    self.face.show_all()
    self.stage.add(self.face)
  def drawhands(self):
    self.hhand=clutter.CairoTexture(30,220)
    self.hhand.set_position(235,250)
    context=self.hhand.cairo_create()
    context.set_operator(cairo.OPERATOR_OVER)
    context.set_source_rgb(1,1,1)
    context.set_line_width(2)
```

Коротко о Clutter

Если вы новичок в *Clutter*, вам, вероятно, будет полезно прочесть первые три урока этой серии (подпишитесь сейчас, чтобы получить бесплатный доступ к PDF через Интернет, или пройдите на стр. 107), а здесь кратко повторим. Сцена *Clutter* – аналог объекта window [окно] на экране. Имеется несколько методов для задания размера, цвета и подключения событий к сцене. Актер – это любой графический объект, появляющийся на сцене. Графические элементы, включая текстовые объекты и текстуры, порождаемые в *Clutter* или заимствованные откуда-либо – это актеры. Они могут изменяться в трех измерениях, и эти преобразования можно анимировать при помощи операций, вызываемых *Clutter*. Обычная процедура – настроить сцену, создать актеров, добавить их на сцену, а затем сделать что-нибудь интересное.

```
    context.move_to(0,0)
    context.line_to(30,0)
    context.line_to(25,120)
    context.line_to(5, 120)
    context.close_path()
    context.stroke_preserve()
    context.set_source_rgb(0,1,0)
    context.fill()
    del context
    self.hhand.show_all()
    self.stage.add(self.hhand)
    # минутная стрелка
    self.mhand=clutter.CairoTexture(16,220)
    self.mhand.set_position(242,250)
    context=self.mhand.cairo_create()
    context.set_operator(cairo.OPERATOR_OVER)
    context.set_source_rgb(1,1,1)
    context.set_line_width(2)
    context.move_to(0,0)
    context.line_to(16,0)
    context.line_to(14,160)
    context.line_to(2, 160)
    context.close_path()
    context.stroke_preserve()
    context.set_source_rgb(0,0,1)
    context.fill()
    del context
    self.mhand.show_all()
    self.stage.add(self.mhand)
    # часовая стрелка
    self.shand=clutter.CairoTexture(9,225)
    self.shand.set_position(246,250)
    context=self.shand.cairo_create()
    context.set_operator(cairo.OPERATOR_OVER)
    context.set_source_rgb(1,1,1)
    context.set_line_width(1)
    context.move_to(0,0)
    context.line_to(9,0)
    context.line_to(5,225)
    context.close_path()
    context.stroke_preserve()
    context.set_source_rgb(1,0,0)
    context.fill()
    del context
    self.shand.show_all()
    self.stage.add(self.shand)
  def update(self, signal):
    z=time.gmtime()
    hangle=30*z.tm_hour+(.5*z.tm_min)-180
    mangle=6*z.tm_min+(.1*z.tm_sec)-180
    sangle=6*z.tm_sec-180
    self.hhand.set_rotation(clutter.Z_AXIS, hangle, 15, 0, 0)
    self.mhand.set_rotation(clutter.Z_AXIS, mangle, 8, 0, 0)
    self.shand.set_rotation(clutter.Z_AXIS, sangle, 5, 0, 0)
if __name__ == '__main__':
  x=clock()
```

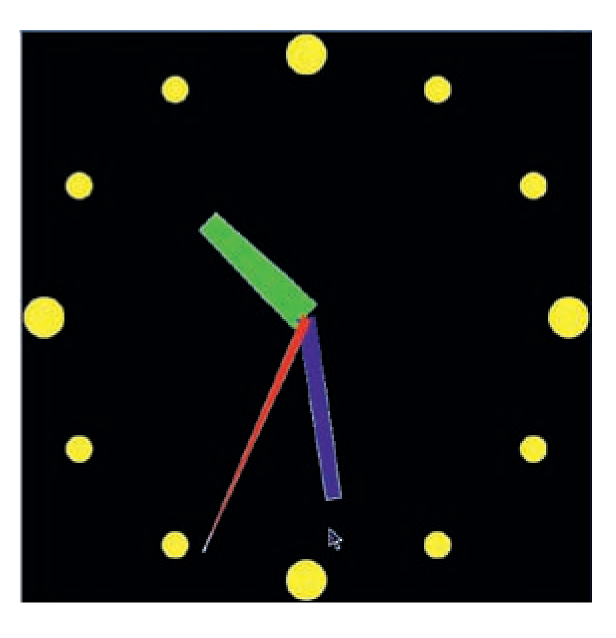
Львиная доля кода уйдет на стрелки часов, ведь для каждой черточки нужна своя линия. Можно сделать текстуры сколь угодно сложными, если, конечно, не жаль времени на написание рисующего их кода.

Изначально мы собирались корректировать системное время при помощи часов NTP в Интернете. Увы, это оказалось малость непрактично – в современных дистрибутивах настроить время из скрипта Python не так-то просто: обычно необходимы привилегии root, а давать их пользовательскому сценарию не здорово! Поэкспериментируем с этой идеей: например, можно периодически получать время с сервера NTP. Шкала времени *Clutter* может использоваться как прерывание, генерируемое, скажем, каждый час. Используя модуль Python *ntplib* (http://pypi.python.org/pypi/ntplib), можно сравнить системное время с «реальным» и получить смещение, которое будет применено к часам после очередного вызова метода **update()**. Вы можете считать свои системные часы очень точными, но они быстро наработают отклонение, особенно на мобильных устройствах. Вот как работает механизм смещения:

```
def ntpcheck(self, signal):
  client=ntplib.NTPClient()
  response=client.request('europe.pool.ntp.org',version=3)
  realtime= time.gmtime(response.tx_time)
  systime=time.gmtime()
  self.offset=(realtime.tm_hour-systime.tm_hour,realtime.tm_
min-systime.tm_min,realtime.tm_sec-systime.tm_sec)
```

Другое возможное дополнение – анимация хода часов. Если честно, они передвигаются на такие короткие расстояния, что многие люди не заметят разницы; но было бы здорово применить одну из «прыгающих» анимаций для эмуляции перемещения стрелки, перескакивающей на новое место, как в старинных часах.

Однако на самом деле наш урок был об использовании объектов *Cairo*. На сайте *Cairo* можно найти куда больше документации о режимах и методах рисования. Вероятно, во многих случаях проще создать текстуру в виде файла и загружать ее при запуске приложения. Но для простых текстур приятно иметь возможность получить приложение, создающее их, и это необходимо для динамических изображений, вроде графиков. LXF



› Наши собственные часы — показывают время в Трамптоне или там, где вы находитесь.

Скорая помощь

Для установки модулей Python существует простой инструмент под названием **easy_install**, обычно это часть пакета **python-setuptools**. Имеется также *Pip*, установщик, который работает как *Yum* или *apt-get*.

Важное замечание о версиях

Наш урок основан на релизе 1.0 библиотеки *Clutter* и 0.98 – модуля *PyClutter*. Более поздние версии поладят с кодом нашего урока, а более ранние – не факт: кроме всего прочего, структура пространства имен в *PyClutter* несколько раз менялась, как и иерархия объектов *Clutter*. Всегда желательно по возможности брать библиотеку *Clutter* и модуль Python из репозитория вашего дистрибутива, если там достаточно новые версии.

» **Через месяц** Создадим великолепные карты при помощи *Clutter* и *Champlain*.

P&B: Резерв

Удивлены загадочной аббревиатурой в заголовке? **Юрий Винник** расшифрует для вас не только ее, но и тонкости процесса резервного копирования.



Наш эксперт

Юрий Винник
Несмотря на обещанный в 2000 году Апокалипсис, стал активно изучать мир свободного ПО. Любит повозится с чем-то новеньким, особенно если это помогает хранить информацию в безопасности.

Перед администратором достаточно большой компьютерной сети рано или поздно (но неизменно остро) встает вопрос о резервном копировании данных. При этом желательно не только сохранять в безопасном месте нажитые непосильным трудом документы пользователей, но и всегда иметь под рукой образы установленных на компьютерах операционных систем, для быстрого восстановления последних в случае отказа.

Ниже мы коснемся двух инструментов для решения этой задачи: специализированного LiveCD для снятия образов разделов по имени PING (PartImage Is Not Ghost) и службы централизованного резервирования *BackupPC*. Их комбинация поможет в максимально сжатые сроки восстановить работоспособность компьютера при крахе файловой системы или даже при физическом выходе из строя винчестера.

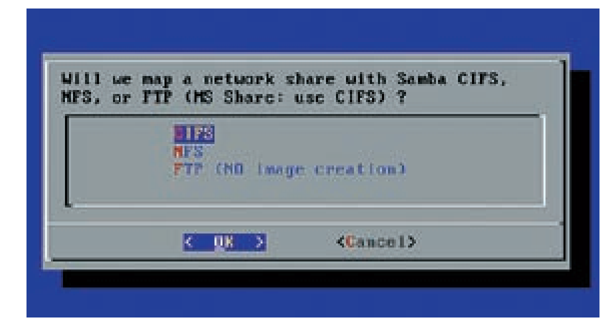
Часть 1: PING не призрак, но Ghost заменит

Скачайте ISO-образ PING с официального сайта (http://ping.windowsdream.com) и прожгите его на диск. Вставьте последний в привод и перезагрузите компьютер (разумеется, не забудьте выбрать CD-ROM в качестве загрузочного устройства в BIOS). После старта дистрибутива вы увидите приглашение, сообщающее, что для входа в интерфейс сохранения/восстановления разделов нужно нажать Enter. Затем система поприветствует нас и напомнит об осторожности, предупредив, что данные образа полностью заместят те, что имеются сейчас на винчестере. Примите это к сведению и нажмите Enter еще раз.

> «PING поддерживает несколько способов создания образов.»

Создаем образ

В появившемся диалоговом окне вам предложат получить административный доступ к системе [Get a shell (root)] или перегрузить/выключить компьютер [Reboot the system/Shutdown]. Выбирайте первый вариант – PING попросит вас указать место для хранения образов дисков, которые вы снимете. Им может быть сетевое устройство [Network Share] или локальный накопитель [Local disk/partition].



› **PING поддерживает распространенные сетевые файловые системы и протокол FTP.**

Скорая помощь

ISO-образ PING (а также многих других дистрибутивов) можно легко превратить в LiveUSB. Для этого воспользуйтесь утилитой *Unetbootin*, доступной по адресу http://unetbootin.sf.net.

Рассмотрим первый случай. После нажатия Enter на Network Share вы почувствуете некую задержку – это система пытается получить сетевые реквизиты через DHCP. Если ей это не удастся, вам придется ввести IP-адрес, маску подсети и шлюз по умолчанию вручную. Далее, надо определиться с протоколом. Им может быть SMB/CIFS (сети Windows), NFS или FTP (только на чтение, то есть с FTP-ресурса можно загрузить образ, но нельзя сохранить снятый). Предположим для определенности, что резервирование выполняется на удаленный Windows-сервер. Создайте на нем новый разделяемый ресурс и защитите его, установив права на чтение-запись только для одного пользователя с паролем. На разделяемом ресурсе должны присутствовать каталоги **Blank_Local_Admin_Passwd** и **Create_New_Image**, иначе PING откажется с ним работать. Это требование касается и двух других протоколов – NFS и FTP, а также Samba, если вы используете CIFS в связке с Unix/Linux.

Теперь у вас всё готово для работы с сетевым хранилищем по протоколу SMB. Чтобы создать образ раздела (или разделов), нужно выполнить несколько простых действий:

1. Выберите в меню пункт CIFS, введите IP-адрес или имя сервера, название разделяемого ресурса, где вы намерены хранить образы, а также имя и пароль допущенного к нему пользователя
2. Укажите раздел (или разделы) жесткого диска, с которых необходимо снять образы. Для множественного выбора используйте клавишу Пробел, подтверждение – Enter.
3. Введите название директории, где следует сохранить образы. Для простоты мы укажем текущий каталог, обозначаемый символом '\'. Дайте образу имя, скажем, XP_02082009.
4. Определитесь со сжатием. *Gzip* работает быстро, но дает меньшую степень компрессии, *bzip2* медленнее, но компактнее. Можно отказаться от сжатия совсем, выбрав пункт No comression.

Дождитесь, пока PING закончит создавать образ выбранного вами раздела. После завершения процесса появится приглашение входа в систему вида «PING login:». PING поддерживает несколь-

» **Пропустили номер?** Узнайте на с. 107, как получить его прямо сейчас.

карман не тянет

ко способов создания образов – рекомендую остановиться на PartImage как на наиболее универсальном. При этом для большинства распространенных типов файловых систем будут сохраняться только занятые блоки диска, что уменьшит место, требуемое для хранения файла образа. В случае NFS все происходит аналогично, разве что имя пользователя и пароль не запрашиваются.

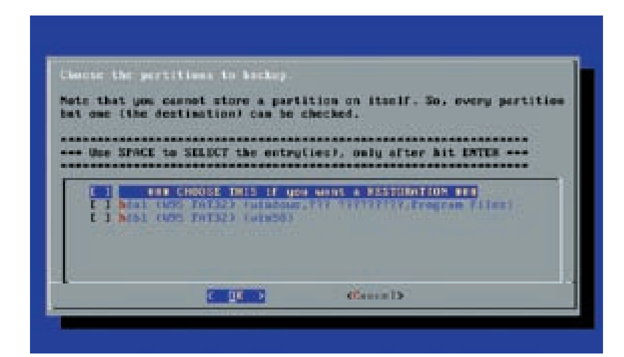
При сохранении образа раздела на локальный диск процедура тоже практически не отличается от описанной. Различие состоит лишь в том, что после указания раздела, подлежащего резервированию, следует ввести не реквизиты удаленного сервера, а имя накопителя, на котором будут сохраняться образы. Им может быть, например, вместительный USB-винчестер.

Восстанови это

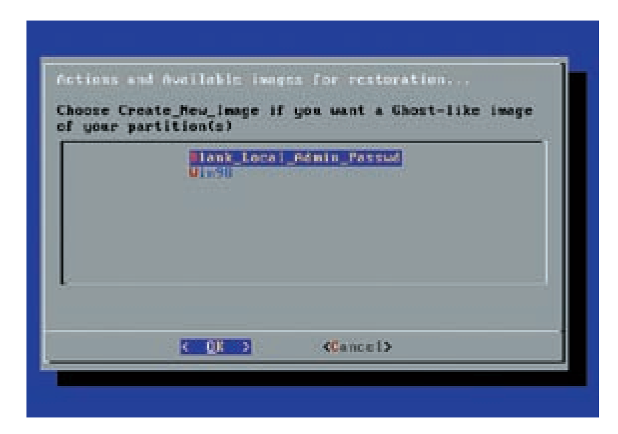
Чтобы восстановить заботливо резервированную вами ОС, нужно снова загрузить PING и выбрать место хранения образа – локальный диск [Local disk/partition] либо сетевой ресурс [Network Share]. Как и раньше, во втором случае потребуется указать протокол доступа к ресурсу (CIFS, NFS либо FTP) и все необходимые реквизиты: сетевой адрес, название, имя пользователя и пароль. Далее необходимо выбрать в меню пункт «###CHOOSE THIS if you want a RESTORATION###» и нажать Enter. Это приведет к появлению меню со списком сохранённых образов – отметьте в нем нужный и нажмите Enter. После удачного восстановления раздела должно появится уже знакомое вам приветствие «PING login:». В случае восстановления с локального диска или раздела потребуется лишь указать каталог, в который смонтировано хранилище образов.

Вставил и забыл

Конечно, восстанавление образов вручную (особенно в случае глобальных катаклизмов в сети) может оказаться делом утомительным. На этот случай разработчики PING предусмотрели возможность создания диска автоматического восстановления, реализующего эту процедуру без участия со стороны администратора. Чтобы сгенерировать его, нужно скачать с сайта разработчиков PING несколько дополнительных файлов: **initrd.gz**, **kernel**,



› Вам будет предложено указать раздел, для которого требуется создать образ.



› Отметьте образ, подлежащий восстановлению, и нажмите Enter.

isolinux.bin, **logo.16**, **boot.msg** и **isolinux.cfg**. Далее, создайте каталог для хранения ISO-образа нового диска (например, **~/MyISO**), а в нём – подкаталог для данных (скажем, **~/MyISO/MyData**), куда следует сохранить эти самые дополнительные файлы и образы разделов компьютера, для которого предназначается «самовосстанавливающийся» диск. Далее, откройте файл **isolinux.cfg** в любом текстовом редакторе (в текущем номере журнала это *Nano*, см. стр. 38), найдите строку, начинающуюся с APPEND, и приведите ее примерно к следующему виду:

```
APPEND vga=normal devfs=nomount Restore_
Only=Y AUTO=Y No_Shell=Y After_Completion=Shutdown
pxe ramdisk_size=50000
load_ramdisk=1 init=/linuxrc prompt_ramdisk=0 initrd=initrd.
gz root=/dev/ram0 rw noapic nolapic lba combined_mode=libata
ide0=noprobe nomce pci=nommconf pci=nomsi irqpoll quiet
```

Параметры должны быть размещены в одну строку, без переносов. Их смысл можно выяснить в документации PING (http://ping.windowsdream.com/ping/doc-2.01/annex.htm).

Теперь всё готово к созданию образа загрузочного диска. Перейдите в **~/MyISO/MyData** и выполните следующую команду:

```
mkisofs -r -b isolinux.bin -boot-info-table -no-emul-boot -boot-
load-size 4 -o ../BurnMe.iso .
```

Обратите внимание на точку в конце команды. Это не опечатка, а указание *mkisofs* искать файлы создаваемого ISO-образа в текущем каталоге.

Сгенерированный загрузочный ISO-образ (**BurnMe.iso**) будет сохранен в директории **~/MyISO**. Его можно прожечь на DVD и использовать для восстановления разделов. Однако будьте внимательны! После загрузки с такого диска восстановление начнется без предупреждения, и вся хранимая на винчестере информация будет потеряна. Если вам нужна интерактивная система восстановления, уберите параметры **AUTO=Y** и **Restore_Only=Y** в строке **APPEND**.

Может статься, что созданные вами образы разделов винчестера превышают объём CD или DVD. Тогда можно создать многодисковый образ. Для начала прикиньте, сколько носителей »

вам для этого понадобится. Далее, создайте в **~/MyISO** столько подкаталогов, сколько у вас будет дисков, например **DVD1**, **DVD2**. Первый диск должен быть загрузочным – скопируйте в него файлы **initrd.gz**, **kernel**, **isolinux.bin**, **logo.16**, **boot.msg** и **isolinux.cfg** и отредактируйте **isolinux.cfg**, как было указано выше. Создайте в этом же подкаталоге пустой файл с именем **MULTI**, без расширения. Кроме этого, скопируйте в **~/MyISO/DVD1** файлы **bios**, **hda**, **hdb**, **hdc**, **hdd**, **sda**, **sdb**, **sdc**, **sdd** и ***.first_sector** из созданного вами образа винчестера. Наконец, заполняйте будущий диск файлами с расширениями ***.001**, ***.002** и так далее, пока на нем не закончится место. Следите только за тем, чтобы эти файлы соответствовали одному разделу винчестера, то есть имели имена типа **hda1.001**, **hda1.002**, … . Второй и последующие диски тоже должны содержать пустой файл с именем **MULTI** и части образов, сформированные по указанному выше принципу. На последнем диске файл **MULTI** должен отсутствовать. Первый диск нужно обязательно сделать загрузочным, как это было показано выше, остальные же можно просто записать на DVD-R как данные.

Часть 2 BackupPC: Сохраним наработанное

Уметь сохранять и восстанавливать образы разделов – это хорошо, но явно недостаточно для создания полноценной системы резервного копирования. У каждого пользователя есть свои данные, которые могут обновляться очень часто. В случае краха раздела или винчестера мы сможем в лучшем случае восстановить их из последнего образа, а это может означать дни и недели потерянного труда. Здесь нам на помощь приходит система автоматического копирования и восстановления пользовательских данных – *BackupPC*.

BackupPC может функционировать только на операционных системах GNU/Linux, BSD или Solaris. Это связано с особенностями файловых систем этих ОС, а именно возможностью создания жёстких ссылок [hard links] на файлы.

Так как *BackupPC* написана на языке Perl, то для её нормальной работы нужно установить некоторые дополнительные Perl-модули, а именно *Compress::Zlib* и *Archive::Zip*, плюс опционально *File::RsyncP*. Кроме того, в системе должен быть установлен web-сервер *Apache* и пакет *smbclient*, если вы хотите использовать для копирования файлов сетевые диски Microsoft Windows или *Samba*; либо *Rsync*, если для этих целей планируется применять одноименный сервис.

Подготовимся

BackupPC, вероятно, доступен в репозиториях вашего дистрибутива, но вы всегда можете загрузить последнюю версию с сайта разработчиков (http://backuppc.sf.net). Сохраните архив в произвольный каталог и распакуйте его. Далее, действуя от имени администратора, выполните установочный сценарий *BackupPC* (**perl configure.pl**) и ответьте на вопросы инсталлятора. Все они хорошо документированы, поэтому обращу внимание лишь на самые важные.

» **BackupPC should run as user [backuppc]** Укажите здесь имя пользователя, от лица которого выполняется *Apache*, поскольку *BackupPC* предоставляет доступ к своему web-интерфейсу только ему.

» **Apache image directory** и **URL for image directory** Здесь указываются полный и относительный пути размещения картинок, необходимых для web-интерфейса. Если в качестве полного пути вы указали, например, **/var/www/BackupPC**, то относительным будет **/BackupPC**.

В подкаталоге **init.d** файла с архивами присутствуют стартовые скрипты *BackupPC* для многих ОС. Скопируйте подходящий в свой **/etc/init.d/** под именем **backuppc**, установите ему права доступа 0755 и добавьте его в автоматически загружаемые системой сервисы.

Наконец, следует изменить владельца каталога, в котором будут хранится резервные копии, на пользователя, от имени которого выполняется *BackupPC* (и, в нашем случае, web-сервер). Используйте команду вроде **chown www-data /var/data**, где **/var/data** – это каталог, в котором будут храниться резервные копии.

Покончив с предварительными настройками, запустите сервис резервного копирования командой:

```
/etc/init.d/backuppc start
```

Web-интерфейс

Перейдем к web-интерфейсу системы. Для начала, назначим администратора (или администраторов) и организуем для них доступ к системе по паролю. Для этого откройте от имени root конфигурационные файлы *BackupPC* и *Apache*: это **/etc/BackupPC/config.pl** и **/etc/apache2/apache2.conf**, соответственно. В первом из них следует найти строку **$Conf{CgiAdminUsers} = ''** и указать в кавычках имена пользователей-администраторов, разделив их пробелами, например: **$Conf{CgiAdminUsers} = 'Andrey Vlad';**. В конфигурационный файл *Apache* нужно добавить следующие строки:

```
<Directory /usr/lib/cgi-bin/BackupPC>
 AuthBasicProvider file
 Options +ExecCGI
 Order deny,allow
 Deny from all
 Allow from localhost
 AuthUserFile /etc/BackupPC/htpasswd
```

› **Введя реквизиты резервируемой системы, не забудьте нажать кнопку Save.**

Main Configuration Editor
Save
Hosts Xfer Email CGI Server Backup Settings Schedule

На стороне клиента

Одним из достоинств *BackupPC* является умение выполнять резервное копирование с клиентов Windows без каких-либо дополнительных программ. Подлежащие резервированию диски и каталоги достаточно просто предоставить в общий доступ стандартными средствами операционной системы. Ресурс должен быть доступен для чтения и записи, и по этой причине его настоятельно рекомендуется ограничить определенным пользователем и паролем. Но так как надёжность сетевых сервисов от Microsoft у вашего автора вызывает большие сомнения, мы пойдем другим путем и будем получать данные с клиентских компьютеров посредством *Rsync*, независимо от платформы. В Linux *Rsync* можно установить через менеджер пакетов, а версию *Rsync* для Windows можно загрузить с сайта *BackupPC* (файл **cygwin-rsyncd-x.y.z_r.zip**). Этот метод требует инсталляции и настройки дополнительного программного обеспечения на компьютере пользователя, но в конечном итоге он стоит усилий.

```
AuthType basic
require valid-user
</Directory>
```

Здесь мы устанавливаем требования для доступа к каталогу **/usr/lib/cgi-bin/BackupPC** (**<Directory /usr/lib/cgi-bin/BackupPC>**), в котором размещён web-интерфейс *BackupPC*, и предписываем использовать реквизиты, сохраненные в файле **/etc/BackupPC/htpasswd** (директива **AuthUserFile /etc/BackupPC/htpasswd**). Естественно, их надо предварительно создать командой **htpasswd**:

```
htpasswd -c /etc/BackupPC/htpasswd Andrey
htpasswd /etc/BackupPC/htpasswd Vlad
```

Проверьте, что все работает, открыв web-интерфейс, расположенный по адресу **http://localhost/cgi-bin/BackupPC/BackupPC_Admin**. После ввода имени пользователя и пароля вы должны увидеть стартовую страницу системы. Если же вы хотите получать доступ к web-интерфейсу *BackupPC* с других компьютеров, перечислите их адреса (можно в виде масок подсетей) в директиве **Allow from** в настройках *Apache*. Теперь, когда ваши хосты готовы к резервному копированию (см. врезку выше), осталось лишь организовать их мониторинг.

Настройка

В первую очередь следует добавить резервируемые системы с помощью пункта меню Правка хостов [Edit Hosts]. На открывшейся странице нужно нажать кнопку Добавить [Add] и ввести доменное имя хоста либо его IP-адрес. При необходимости, если компьютер получает имя с DHCP-сервера, нужно отметить галочкой пункт «dhcp». Все действия нужно сохранять нажатием кнопки Сохранить [Save] вверху страницы.

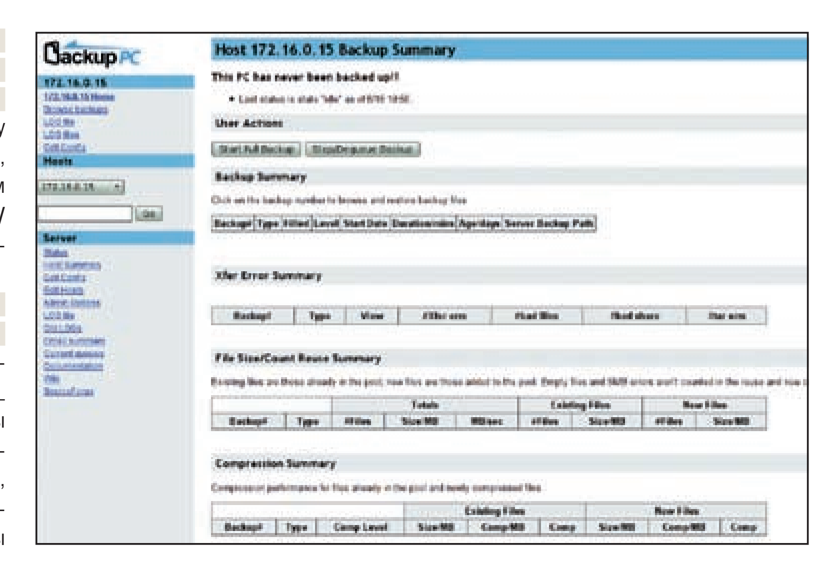
После добавления всех резервируемых хостов можно указать дополнительные параметры либо для системы в целом, через пункт меню Правка конфигурации [Edit Config], либо индивидуально для каждого хоста, выбрав из выпадающего списка Укажите хост...[Select a host...] интересующую нас систему, а затем щёлкнув на пункте Edit Config в меню для выбранного хоста. Давайте рассмотрим наиболее важные из доступных опций.

> «Система проверяет наличие изменений в файлах каждый час.»

Параметр XferMethod на вкладке Передача [Xfer] задаёт метод доступа к резервируемым данным. Укажите здесь smb, если планируете получать к ним доступ по протоколу SMB/CIFS (то есть если вы пошли первым путём и просто экспортировали подлежащие резервированию каталоги как разделяемые ресурсы Windows), или rsyncd, если используется сервис *Rsync*. Также на этой же вкладке нужно обязательно указать имя сетевого ресурса SmbShareName (либо RsyncShareName). Если сетевых ресурсов на одном хосте несколько, их можно добавить с помощью кнопки Add. Ещё потребуется указать имя пользователя, которому разрешён доступ к данному ресурсу (SmbShareUserName), и его пароль (SmbSharePasswd). Для сервиса *Rsync* эти параметры имеют название RsyncdUserName и RsyncdPasswd соответственно.

В глобальных настройках системы, Сервер > Правка конфигурации [Server > Edit Config], кроме указанных выше параметров, можно задать ещё некоторые, влияющие на систему *BackupPC* в целом. Так, WakeupSchedule на странице Server регулирует расписание, по которому система проверяет наличие изменений в файлах резервируемых систем. По умолчанию, проверка проводится каждый час, кроме полуночи.

После изменения глобальных настроек нужно перезагрузить конфигурацию серевера. Делается это путём нажатия кнопки Перезагрузить [Reload], которую можно найти в Сервер > Опции администратора [Server > Admin Options].



› Пока что здесь пусто, но скоро начнёт накапливаться информация о резервируемых данных.

Все вместе

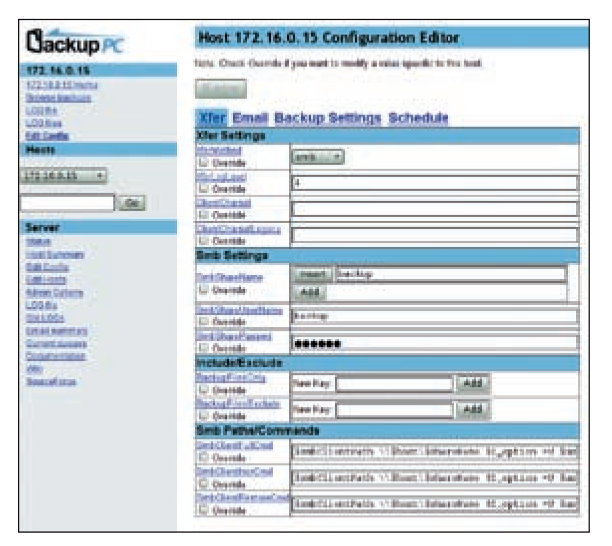
Теперь, чтобы обеспечить резервирование данных пользователя, нужно лишь:

1. Открыть доступ к резервируемым ресурсам по сети, либо установить сервис *Rsync* и описать ресурсы через него.
2. Создать образ разделов винчестера на сетевом ресурсе либо сохранить их на CD/DVD с помощью дистрибутива PING;.
3. Добавить данные о резервируемых ресурсах в систему BackupPC.

Для быстрого восстановления работоспособности компьютера в случае сбоя будет достаточно:

1. Загрузить дистрибутив PING и восстановить разделы.
2. Загрузить восстановленную систему, запустить браузер, войти в web-интерфейс *BackupPC*, выбрать в меню Hosts [Хосты] восстанавливаемую систему и, далее, отметить последнюю резервную копию в пункте Backup Summary [Сводка резервирования] на открывшейся странице и восстановить из неё все пользовательские файлы.

Если же у пользователя просто произошла потеря либо повреждение файлов, без краха разделов, то восстановить данные можно, даже не подходя к его компьютеру. Для этого достаточно открыть web-интерфейс и отметить подлежащие восстановлению файлы, не вставая со своего рабочего места. По-моему, удобно. LXF



› Для каждого введённого в систему компьютера можно указать свои индивидуальные настройки.

RAID-5: Защита

Эксперт по дисковым травмам **Бен Мартин** наделит вас мощью программного RAID, чтобы ваши данные пережили смерть винчестера.



Наш эксперт

Бен Мартин работает с файловыми системами более десяти лет. После получения ученой степени он основал консультационное агентство по *libferris*, файловым системам, поисковым решениям и разработке на C++/*Qt*.

В наши дни ядро Linux обеспечивает исключительно хорошую программную поддержку RAID-массивов (Redundant Array of Inexpensive Disks [избыточный массив независимых/недорогих жёстких дисков]). Достаточно подключить нужное число накопителей и сказать ядру, как настроить ваш RAID. Вы можете назначать требуемую избыточность: скажем, нужна защита от потери двух дисков из пяти без повреждения данных. Тут есть и оборотная сторона: чем больше дисков вы можете позволить потерять, тем меньше полезного места будет на вашем RAID.

Многие материнские платы объявляют, что предоставляют аппаратный RAID. BIOS нередко реализует его через системный процессор, так что по сути этот RAID тоже программный. Основной повод выбрать для построения RAID ядро Linux вместо прошивки на материнской плате – это свобода переноса RAID-массива на другую машину при обновлении вашего оборудования.

Программный RAID в ядре Linux включает различные его типы: зеркалирование, чередование, RAID с контролем четности (RAID-5 и RAID-6) и RAID-10 (комбинация зеркалирования и чередования). Ядро также более гибко – к примеру, если вы хотите добавить новый 1-ТБ диск в RAID, сделанный вами шесть месяцев назад, можно расширить старый RAID, включив в него новый диск. Скорее всего, придется выключить питание компьютера, чтобы добавить само устройство, но ядро может встроить новый диск в RAID, даже когда вы пользуетесь машиной. Различные конфигурации RAID имеют различные параметры подмены одних дисков другими – сколько и каких дисков может «умереть», не вызвав потери ваших данных, много ли дискового пространства вы реально сможете использовать и как быстр будет ваш RAID. Например, массив с чередованием (RAID-0) пишет блоки данных на оба диска по очереди и, следовательно, достаточно быстр. Оборотной же стороной является то, что в RAID-0 при гибели любого диска вы потеряете все свои данные.

Создавая RAID с помощью ядра Linux, вы сообщаете ему, какие разделы на каких дисках объединить в массив. Затем вы получаете новое RAID-устройство – скажем, **/dev/md/md-что-то**, и можете использовать **md-что-то** как любой другой раздел – например, запустить **mkfs.ext3 /dev/md/md-что-то** для создания на нем файловой системы ext3 и затем смонтировать ее куда-нибудь.

Правда, минус данного подхода – необходимость создания нового RAID для каждой файловой системы, которая вам потребуется: захотите, чтобы **/usr** и **/tmp** были на разных файловых системах – создавайте два RAID-устройства. Удобно будет использовать на RAID-устройстве логическое управление томами [Logical Volume Management, LVM] и затем создать множество различных файловых систем на одном и том же RAID-массиве. LVM также позволяет использовать для настройки файловых систем инструменты с графическим интерфейсом, такие как *system-config-lvm*. Естественно, все это можно проделывать и через командную строку. Еще один повод использовать LVM поверх RAID – возможность расширения и миграции логического тома. Если с годами потребуется настроить новый RAID, можно будет просто перенести ваш логический том со старого RAID на новый.

Наш урок научит вас настраивать программный RAID и ставить на него LVM для управления разделами. Так как графический инструмент *system-config-lvm* достаточно дружелюбен к пользователю, процесс разбиения с использованием LVM будет рассмотрен не очень подробно. Мы сфокусируемся на установке RAID и *mdadm*. Я также расскажу, как добавить в RAID новый жесткий диск, чтобы вы смогли со временем расширить его (включая и LVM). Рекомендуется как минимум прочесть страницу руководства *mdadm*, чтобы чувствовать себя комфортно с описанной процедурой. За целостность и доступность ваших данных отвечаете вы и только вы.

Детали RAID

Две наиболее интересные для вас группы RAID – RAID-10 и «RAID с четностью»: RAID-5 или RAID-6. Если у вас шесть терабайтных дисков, RAID-10 даст вам 3 ТБ дискового пространства (половина объема вам не доступна: она используется для защиты данных от повреждения). Зато вы можете позволить себе потерять до трех дисков в массиве без угрозы потери данных.

» **Месяц назад** Мы изучили быструю виртуализацию с OpenVZ.

вашИх данных

С другой стороны, в RAID-5 нужен отдельный диск для хранения контрольных сумм, а остальные можно использовать полностью, получив при той же схеме 5 ТБ дискового пространства. RAID-6 похож на RAID-5, с тем исключением, что для контрольных сумм вам потребуются два диска. В RAID с проверкой четности число дисков контрольных сумм равно числу дисков, которых вы можете лишиться без потери данных: 1 для RAID-5 или 2 для RAID-6.

При потере диска вы можете просто заменить его на новый и велеть системе записать на него данные и контрольные суммы для восстановления избыточности, чтобы опять позволить себе терять один диск, сохраняя данные.

Я отметил, что для RAID-10 можно потерять до трех дисков из шести без потери данных. «До трех» означает, что это число зависит от того, какие диски вы потеряете. Так как RAID-10 – комбинация чередования и зеркалирования, и каждый конкретный блок данных пишется на на два диска (зеркалирование), то потеря обоих из этих двух дисков уничтожит и ваши данные! Это условная вероятность, которую защитники RAID-10 часто упускают из виду.

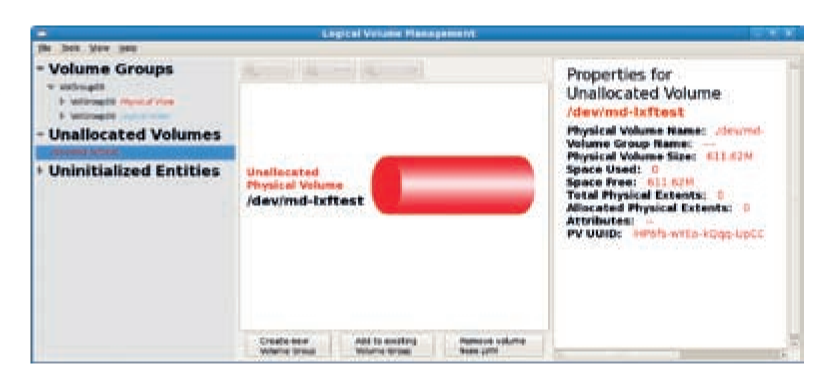
Для RAID с контрольными суммами вам также следует применять источники бесперебойного питания (UPS), потому что запись на RAID при сбое питания может повредить его целиком или частично [это более вероятно, чем может вам показаться, – *прим. ред.*]. Допустим, вы записали данные на один из пяти дисков, но диск с контрольными суммами перед сбоем записаться не успел. В этом случае RAID не будет знать, чему верить – пяти дискам с данными или информации на диске с контрольными суммами.

Настраиваем RAID

Для настройки LVM поверх RAID-5 или RAID-6 применяются два различных инструмента: *mdadm* и какая-либо из утилит LVM, например, *system-config-lvm.* Сперва вы создаете RAID-устройство, которое объединяет вместе набор разделов и реализует функциональность RAID.

Наши тесты выполнялись с использованием небольших дисков внутри виртуальной машины. Эти диски отображаются в виртуальной машине как обычные, вида **/dev/sdb**, и в целях нашего урока функционируют как настоящие физические диски. Я также настраивал много настоящих массивов RAID-5, RAID-6 и RAID-10 в различных конфигурациях с использованием реальных жестких дисков от сотен мегабайт до терабайт объема.

Тестовая установка использует четыре 200-МБ виртуальных диска, которые выглядят как **/dev/sdc**, **sdd**, **sde** и **sdf**. Отличие между реальным физическим и виртуальным диском в том, что виртуальный диск не появляется, как он делал бы на реальной машине, в директории **/dev/disk/by-id**. Директория **/dev/disk/by-id** отображает список дисков по изготовителю, модели и серийному номеру. Каждый раздел на каждом из этих дисков имеет свою ссылку. Я бы рекомендовал использовать ссылки в **/dev/disk/by-id** при создании RAID с помощью *mdadm*, потому что именование физических дисков по изготовителю, модели и серийному номеру оставляет меньше простора для человеческого фактора, приводящего к ошибкам из-за указания неверного диска. Если вы используете новые диски и планируете задействовать их в RAID целиком, я все равно предлагаю создать на каждом из новых дисков одиночный раздел: тогда в будущем, добавляя новый диск для расширения RAID, вы будете иметь лишний способ определить, что это именно новый диск, так как на нем не будет раздела. Чтобы создать такой раздел, запустите *fdisk* на каждом новом диске, наберите N (новый), P (основной), 1 и дважды нажмите Enter, чтобы использовать весь диск. Измените t (тип) раздела на FD (Linux raid autodetect) и W (запишите) таблицу разделов. Вот пример таблицы разделов, настроенной таким образом:



› RAID наконец появился в *system-config-lvm*.

```
# [id:fdisk-sdc]
# fdisk -l /dev/sdc
Disk /dev/sdc: 214 MB, 214748160 bytes
64 heads, 32 sectors/track, 204 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes
Disk identifier: 0x959ddf6b
Device    Start   End ... Id  System
/dev/sdc1     1   204 ... fd  Linux raid autodetect
```

При создании RAID будет полезным дать RAID-устройству более осмысленное имя, чем идущее по умолчанию – вида **/dev/md/md0**. Это не обязательно, но облегчит вам поиск, если вы используете несколько RAID-массивов. Если вы измените имя устройства, затем нужно использовать для *mdadm* параметр командной строки **--auto**, чтобы сказать ему, RAID-устройство какого типа он должен для вас создать.

Команда, приведенная в следующем примере, создает новый RAID-5, используя четыре выбранных диска, и задает приятное «говорящее» имя для RAID-устройства. Создав RAID, я проверил параметры его присутствия и увидел, что состояние RAID – «чистый» [clean], и каждый из четырех дисков активен.

На беду, *mdadm* выводит имена разделов, используя соглашение именования **/dev/sdX**, и приходится рыться в **/dev/disk/by-id**, выясняя, к какому диску с каким серийным номером относится сообщение *mdadm*. Хотя *mdadm* рапортует о дисках RAID, используя **/dev/sdc**, реально буква диска может меняться от загрузки к загрузке, и *mdadm* должен продолжать правильно находить и настраивать RAID. К примеру, я вначале создал диск в **/dev/sdb**, который не содержал данных и существовал исключительно для того, чтобы первый раздел RAID был на **/dev/sdc**. После создания и установки RAID я удалил **/dev/sdb** из виртуальной машины и перезапустил ее.

RAID продолжал правильно работать, хотя после перезагрузки четыре диска изменили свою позицию (старый **sdc** стал теперь **sdb**, и т. д.). Этого можно было ожидать, но проверка никогда не помешает.

```
# [id:mdadm-create]
# Creating a RAID device for four disks
```

»

```
# mdadm --create --auto=md --verbose \
  /dev/md/md-lxftest --level=5 --raid-devices=4 \
  /dev/sdc1 /dev/sdd1 /dev/sde1 /dev/sdf1
# mdadm --detail  /dev/md/md-lxftest
/dev/md/md-lxftest:
      Version : 00.90.03
  Creation Time : Thu Feb 11 22:50:49 2010
    Raid Level : raid5
    Array Size : 628800 (614.17 MiB 643.89 MB)
  Used Dev Size : 209600 (204.72 MiB 214.63 MB)
   Raid Devices : 4
  Total Devices : 4
Preferred Minor : 127
   Persistence : Superblock is persistent
   Update Time : Thu Feb 11 22:51:02 2010
         State : clean
 Active Devices : 4
Working Devices : 4
 Failed Devices : 0
  Spare Devices : 0
        Layout : left-symmetric
    Chunk Size : 64K
          UUID : 470970a6:d264c64e:c3926...cf805a
        Events : 0.4
Number Major Minor  RaidDev State
   0    8    32    0   active sync   /dev/sdc1
   1    8    48    1   active sync   /dev/sdd1
   2    8    64    2   active sync   /dev/sde1
   3    8    80    3   active sync   /dev/sdf1
```

Если при создании RAID вы допустили ошибку, вам, вероятно, захочется удалить его и начать сначала. Команда для избавления от RAID показана в следующем блоке кода. Если у вас настроен LVM поверх RAID, вам, скорее всего, придется сперва удалить все логические разделы и все физические тома, созданные ранее на RAID. Причина, по которой я перезагрузился, была в необходимости убедиться, что процедура загрузки не попытается восстановить RAID. Это помогает предотвратить попытку заново использовать диски из только что удаленного RAID.

Если вы удалили и пересоздали RAID, используя те же разделы, вы будете предупреждены, что это устройство уже было частью массива RAID, чтобы вы случайно не добавили в новый RAID неправильный диск. Если вы продолжите создавать RAID, вы затем должны запуститься с состоянием «clean, degraded, recovering», и один из дисков будет в состоянии «свободен-перестроение» (spare rebuilding). При большом объеме используемых дисков процесс перестройки может занять немалое время – так, для 500-ГБ дисков потребуется несколько часов. А вот как удаляется RAID-массив:

```
# [id:mdadm-delete]
# vi /etc/mdadm.conf
... удалите запись, если она есть ...
# mdadm --manage --stop /dev/md/md-lxftest
# sync
# reboot
```

Загрузочные сценарии Fedora активируют RAID при старте, только если существует файл **/etc/mdadm.conf**. Если это единственный созданный вами RAID, то команда из приведенного ниже кода (**id:mdadm-config**) создаст корректный конфигурационный файл. Данной спецификации может соответствовать множество устройств – это заставит *mdadm* просматривать все ваши разделы, чтобы попытаться найти используемые в вашем RAID-массиве. Устройства также могут быть указаны посредством шаблонов. Если вы создали RAID, задействовав первые разделы четырех дисков, которые имеют одни и те же название изготовителя и размер, то применение пути **/dev/disk/by-id/** и маски поможет выбрать только интересующие нас диски и разделы, сэкономив время на поиске во время загрузки. К примеру, спецификация вида **DEVICE /dev/disk/by-id/ata-SAMSUNG_HD501LJ*part1** велит *mdadm* проверять только первые разделы в коллекции 500-ГБ дисков Samsung. Если вы настраиваете новый RAID, то можете добавить второй раздел **DEVICE**, приказав *mdadm* также просматривать эти диски для поиска разделов RAID.

```
# [id:mdadm-config]
# echo 'DEVICE partitions' >> /etc/mdadm.conf
# mdadm --detail --scan    >> /etc/mdadm.conf
# cat /etc/mdadm.conf
DEVICE partitions
ARRAY /dev/md/md127 level=raid5 num-devices=4
UUID=eb..
```

Хотя созданная выше конфигурация должна заработать, вы можете решить изменить строку **ARRAY**, сгенерированную запущенным *mdadm*. В частности, полезно бывает использовать специальные имена устройств и удалить параметр **num-device**. Удаление аргумента **num-device** означает, что в будущем при расширении RAID-массива вам не придется помнить о необходимости подправить это значение. Если вы изменяете файл устройства, добавьте аргумент auto, чтобы сказать *mdadm*, какой тип устройства нужно создать для вас.

Окончательная конфигурация будет такой:

```
# [id:/etc/mdadm.conf]
DEVICE partitions
ARRAY /dev/md/md-lxftest auto=md level=raid5 UUID..
```

Расслоение LVM

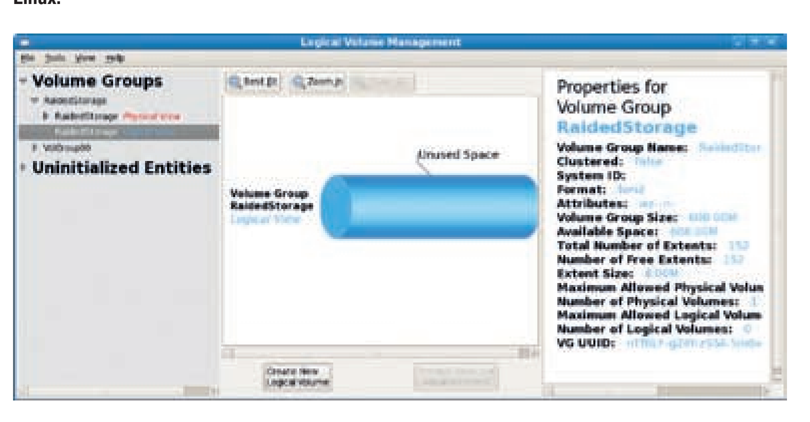
Новое RAID-устройство, которое мы только что создали, не будет немедленно доступно из *system-config-lvm*. Вы должны запустить на нем *pvcreate*, чтобы оно стало видимым, как показано в коде ниже (**id:pvcreate-for-md**). На этом этапе вы увидите новый неразмеченный том для вашего RAID, как представлено на экранном снимке слева.

Теперь вы можете использовать графический интерфейс *system-config-lvm* для создания новой группы томов и настройки нескольких логических томов на RAID для использования с вашими файловыми системами. Кликните на «Create New Volume Group» [Создать новую группу томов] для выделения новой группе томов свободного физического тома, созданного в (**id:pvcreate-for-md**). Создать новую группу томов достаточно просто. Выбрав новую группу томов, откройте дочернее 'Logical View' [Логическое представление] этой группы, и вы увидите пункт 'Create New Logical Volume' [Создать новый логический том]. На этом этапе вы заметите опцию создания новых логических томов внутри логического отображения группы томов. Именно на логических томах в LVM вы и создаете файловые системы Linux. Поэтому создание нового логического тома также позволит вам установить файловую систему и точку монтирования.

Скорая помощь

Более подробно об LVM см. на www.tldp.org/HOWTO/LVM-HOWTO/index.html.

›Логическое отображение новой группы томов: мы можем теперь сделать раздел для файловой системы Linux.



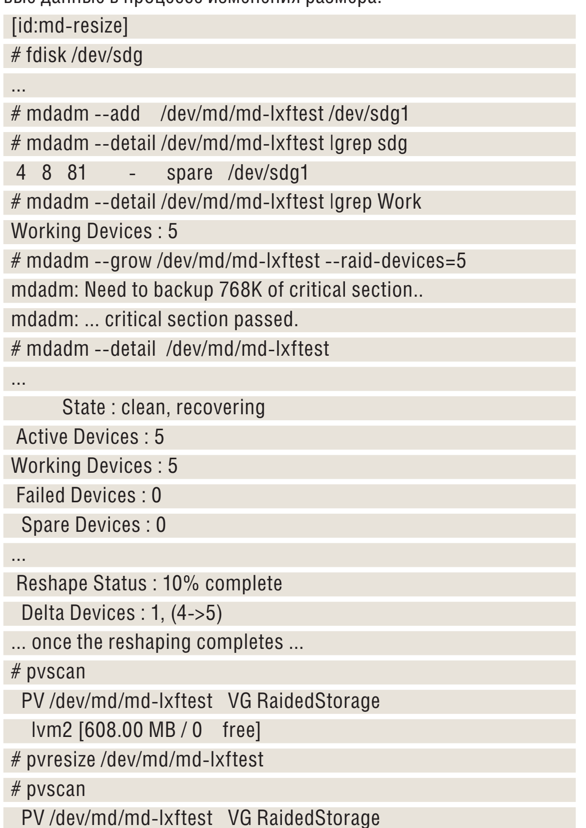
```
# [id:pvcreate-for-md]
pvcreate /dev/md/md-lxftest
```

Расширяем RAID

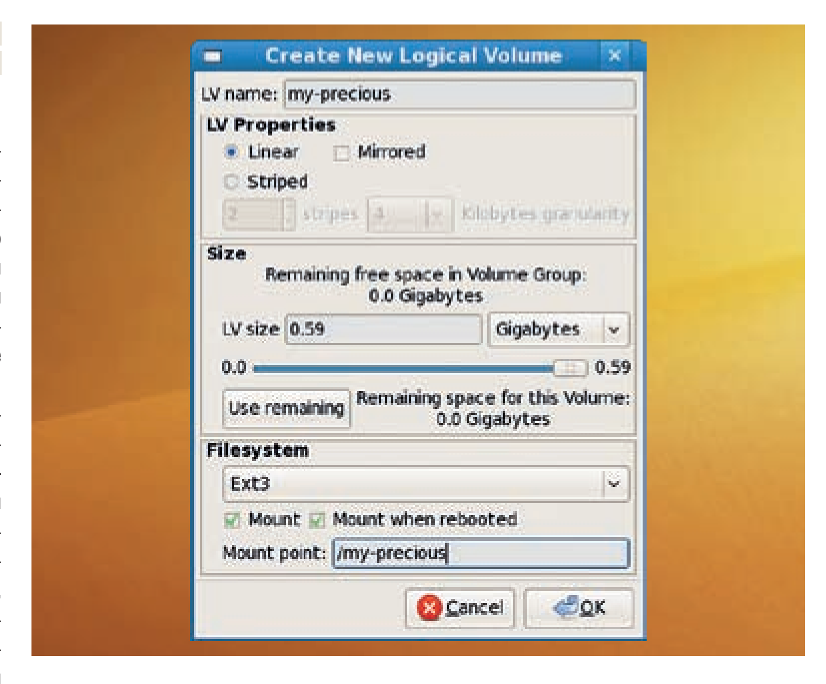
Чтобы убедиться, что RAID можно расширить без потери содержащихся на нем данных, я скопировал на массив достаточно большой файл и записал его сумму MD5. Замечу что в оригинале логический том для **/my-precious** занимал все доступное пространство на группе томов RAID. При расширении файловой системы я велел *lvextend* использовать все доступное пространство при изменении размера логического тома. Это, пожалуй, наиболее часто используемый сценарий: вы добавляете новый диск в RAID и расширяете раздел, который стал тесен для данных.

Новый диск, добавляемый в RAID – это **/dev/sdg**. Как и в случае других дисков, я сперва создал одиночный раздел и установил его тип в FD. Команды и часть их вывода показаны в (**id:md-resize**). После добавления нового диска в RAID-массив он был помечен как свободный (spare). Команда **mdadm --grow** велит массиву изменить свой размер для включения нового дискового раздела. Для RAID-5 это потребует большой дисковой активности, так как новый диск будет вовлечен в процедуру контроля четности, и блоки потребуется перераспределить. По завершении переразметки *pvresize* включит новое пространство в физический том LVM. Команда *lvextend* расширит логический том LVM, заняв назначенное пространство (в данном случае я велел ей использовать все свободное пространство, доступное в физическом томе). Сама файловая система все еще будет иметь старый размер. Финальным шагом будет использовать **resize2fs** или **xfs_growfs** (или аналогичный инструмент для конкретной файловой системы), чтобы изменить размер файловой системы и сделать новое пространство доступным.

Я оставил файловую систему **/my-precious** смонтированной в течении всей операции, хотя на самом деле не писал в нее новые данные в процессе изменения размера.



```
[id:md-resize]
# fdisk /dev/sdg
...
# mdadm --add   /dev/md/md-lxftest /dev/sdg1
# mdadm --detail /dev/md/md-lxftest |grep sdg
 4  8  81      -     spare   /dev/sdg1
# mdadm --detail /dev/md/md-lxftest |grep Work
Working Devices : 5
# mdadm --grow /dev/md/md-lxftest --raid-devices=5
mdadm: Need to backup 768K of critical section..
mdadm: ... critical section passed.
# mdadm --detail  /dev/md/md-lxftest
...
        State : clean, recovering
 Active Devices : 5
Working Devices : 5
 Failed Devices : 0
  Spare Devices : 0
...
 Reshape Status : 10% complete
  Delta Devices : 1, (4->5)
... once the reshaping completes ...
# pvscan
  PV /dev/md/md-lxftest   VG RaidedStorage
    lvm2 [608.00 MB / 0    free]
# pvresize /dev/md/md-lxftest
# pvscan
  PV /dev/md/md-lxftest   VG RaidedStorage
    lvm2 [812.00 MB / 204.00 MB free]
# df -h  /my-precious
Filesystem          Size  Used Avail Use% Mounted on
/dev/mapper/RaidedStorage-my--precious
                    599M  543M   26M  96% /my-precious
# lvextend -l '+100%FREE' \
    /dev/RaidedStorage/my-precious
  Extending logical volume my-precious to 812.00 MB
  Logical volume my-precious successfully resized
# df -h /my-precious/
Filesystem          Size  Used Avail Use% Mounted on
/dev/mapper/RaidedStorage-my--precious
                    599M  543M   26M  96% /my-precious
# resize2fs /dev/RaidedStorage/my-precious
resize2fs 1.40.2 (12-Jul-2007)
Filesystem at /dev/RaidedStorage/my-precious
  is mounted on /my-precious;
  on-line resizing required
old desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of
  /dev/RaidedStorage/my-precious
  to 207872 (4k) blocks.
The filesystem on /dev/RaidedStorage/my-precious
  is now 207872 blocks long.
# df -h /my-precious/
Filesystem          Size  Used Avail Use% Mounted on
/dev/mapper/RaidedStorage-my--precious
                    799M  543M  218M  72% /my-precious
# md5sum /my-precious/largefile
... порядок! ...
```



› **Новый логический том для моих драгоценных файлов.**

С учетом низкой цены 1,5-ТБ дисков по состоянию на февраль 2010, самое время соединить четыре таких накопителя в RAID-5 или шесть дисков в RAID-6. Если у вас не хватает портов SATA, недорогая PCI-карта Silicon Image 3114 даст вам четыре SATA-порта, готовых к новому четырехдисковому RAID-5.

Сгруппировав новые диски вместе в ваш RAID и создав на этом массиве физический том LVM, вы сможете использовать *system-config-lvm* обычным образом, но ваши данные будут защищены от аппаратных сбоев. LXF



Более подробно о программном RAID см. на http://tldp.org/HOWTO/Software-RAID-HOWTO.html.

» **Через месяц** Подцепим календарь Google к вашему телефону через SMS.





Есть вопрос по открытому ПО? Пишите нам по адресу answers@linuxformat.ru

В этом месяце мы ответим на вопросы про:

1 Ubuntu и 3G
2 Каталоги Windows в общем доступе
3 Мониторинг сервера
4 Печать и Web-браузеры
5 Проблемные разделы
6 Пропавший жесткий диск USB
7 Linux на PS3
8 Извещения Google
9 Снова про печать
10 Grub 2
11 NetworkManager

1 3G-проблемы Ubuntu

В У меня серьезный вопрос про DVD LXF127. Даже если вы не согласитесь с моим предложением — пожалуйста, хотя бы поясните. Я без проблем пользовался 3G-соединением в Ubuntu 9.04 — правда, иногда требовалось переподключить модем, чтобы появилась связь. А затем я установил Ubuntu 9.10, и Интернет исчез — оказалось, это известная проблема. Остается вернуться к Ubuntu 9.04. Если я воспользуюсь вашим DVD, я что, тоже лишусь Интернет-соединения? Или же ядро, решающее мою проблему, туда включено?

Отсюда и мое предложение. Раз 3G-соединения создают проблемы, пожалуйста, оговаривайте, сложно ли неопытному пользователю подключиться к Интернету через 3G-модем. Ведь иначе будет много разочарований. В одной только Финляндии сотни тысяч пользователей уже имеют соединения 3G. Странно, что Canonical выпустила релиз 9.10, зная о данной проблеме. Ведь этак они потеряют массу потенциальных пользователей!

Боб Эбспел [Bob Abspoel]

О Да, похоже, для Ubuntu 9.10 (включая тот, что на LXFDVD) эта проблема довольно типична. Впрочем, когда Вы прочтете наш ответ, ее, скорее всего, уже решат. Если 3G – не единственный Ваш способ выхода в Сеть, установите Ubuntu 9.10, подключитесь к Интернету через Ethernet или Wi-Fi, а затем скачайте все доступные обновления. В противном случае, соединение 3G в *Network Manager* можно довольно легко «оживить», хотя придется выйти в командную строку. Сначала определите идентификаторы Product ID и Vendor ID для Вашего модема, воспользовавшись командой **lsusb**. Выглядит это примерно так:

```
sudo lsusb
bus 005 Device 008: ID 12d1:1003 Huawei Technologies Co., Ltd. E220 HSDPA Modem / E270 HSDPA/HSUPA Modem
```

Здесь **12d1** – Vendor ID, а **1003** – Product ID. Теперь скомандуйте

```
gksudo gedit /etc/modprobe.d/usbserial
```

Добавьте в файл следующую строку, подставив полученные значения ID:

```
options usbserial vendor=0x12d1 product=0x1003
```

Перезагрузитесь, и при подключении устройства оно должно обнаружиться. Если нет, зайдите в меню Переходы > Мой компьютер [Places > Computer] и посмотрите, не распознан ли модем как привод CD-ROM; тогда выберите опцию Извлечь [Safely Remove]. Это якобы «устройство» – виртуальный CD во флэш-памяти модема, содержащий ПО и ненужные Вам драйверы Windows. Данный шаг можно автоматизировать, дав команду

```
gksudo gedit /etc/udev/rules.d/10-local.rules
```

и приписав в файле правил следующую строку:

```
ATTRS{vendor}=="HUAWEI",
DRIVERS=="sr", OPTIONS="ignore_device"
```

Здесь предполагается, что Ваш модем – Huawei (он самый распространенный). Данное правило велит системе игнорировать устройства, созданные Huawei и использующие драйвер **sr**. А драйвер применяется для CD и DVD, поэтому система будет игнорировать виртуальные CD, но не модем. **MC**

2 Доступ к сети Windows

В У меня система с двойной загрузкой — Karmic и Windows 7, плюс еще два компьютера в сети, работающих под различными версиями Windows. Вся информация хранится в сети, и она доступна с любого компьютера, даже при работе в *VirtualBox* из дистрибутива Linux.

Я думал, что уже созрел до отказа от Windows, но тут возникла проблема, которую я решить не смог. Выбрав Переходы [Places], я получаю полный список всех дисков на компьютере Linux, плюс ссылку на сеть, через которую можно получить доступ к дискам других компьютеров.

Но при попытке открыть файлы через меню Открыть [Open] в любой программе ссылка на сеть не появляется. Я упустил какую-то настройку, или нужно добавить еще какой-нибудь код?

Кэс Стефенсон [Cas Stephenson]

О Обзор сети браузером, созданным для поиска сетевых ресурсов и доступа к ним, и открытие файла по запросу из обычных приложений для работы с иерархией файлов на компьютере, действительно, различаются. Открывать файлы через сеть умеют далеко не все программы. Решение – монтировать сетевые ресурсы так, чтобы они отображались в составе каталогов Вашей локальной файловой системы. Каталог, к которому Вы монтируете се-

Наши эксперты

» Мы найдем ответы на любой вопрос — от проблем с установкой системы или модемом до сетевого администрирования; главное — спросить!



Нейл Ботвик
Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



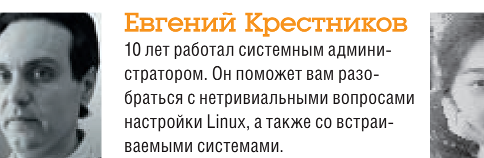
Майк Сондерс
Майк был одним из создателей прототипа LXF – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.



Пол Хадсон
Пол – местный супер-программист, и он может и хочет управиться со всеми вашими проблемами по части web и баз данных.



Грэм Моррисон
Когда Грэм не обозревает кучи программного обеспечения и не халтурит с MythTV, он готов к ответам насчет любого оборудования и проблем виртуализации.



Евгений Крестников
10 лет работал системным администратором. Он поможет вам разобраться с нетривиальными вопросами настройки Linux, а также со встраиваемыми системами.



Юлия Дронова
Если компьютер у Юлии не занят выполнением команды *emerge*, она спешит применить его для модерирования www.unixforum.org.

КУДА ПОСЫЛАТЬ ВОПРОСЫ

Пишите нам по адресу: answers@linuxformat.ru **или спрашивайте на форуме:** www.linuxformat.ru

тевой ресурс, называется точкой монтирования [mount point]. Он должен уже существовать в Вашей локальной файловой системе. Он не обязан быть пустым, но пока к нему примонтирована сетевая файловая система, его содержимое все равно невидимо, так что пустота практичнее.

Раз Вы упомянули сетевой браузер Ubuntu, то в своих Linux-системах Вы, скорее всего, применяете *Samba* для доступа к каталогам, использующим систему Windows CIFS, как и в Windows, конечно. Чтобы монтировать сетевые ресурсы Windows в Ubuntu, необходимо установить пакет **smbfs**. После этого можно примонтировать нужный ресурс следующей командной строкой:

```
sudo mount -t cifs //server/share /mount/point -o guest
```

Убедитесь, что Вы подставили правильное имя сетевого компьютера и размещенного на нем разделяемого ресурса. Если для доступа к ресурсу требуется аутентификация, то в командной строке можно указать имя пользователя и пароль, например:

```
sudo mount -t cifs //server/share /mount/point -o username=xxx,password=yyy
```

Но такой подход небезопасен. Команда с явным указанием пароля будет работать все время, пока примонтирован ресурс – и любой пользователь сможет узнать Ваш пароль, воспользовавшись утилитой **ps**. Чтобы решить эту проблему, укажите данные аутентификации в файле, в форме

```
username=xxx
password=yyy
```

И дайте команду **mount**, указав имя этого файла:

```
sudo mount -t cifs //server/share /mount/point -o credentials=/path/to/file
```

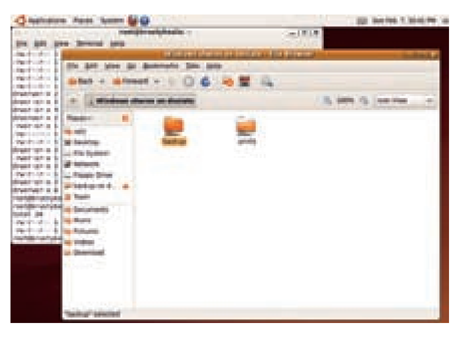
Когда все заработает, Вы, возможно, предпочтете автоматизировать данный процесс. Для этого поместите файл с данными аутентификации в каталог **/etc/samba** и дайте пользователю root (и только ему!) доступ к этому файлу с правом чтения:

```
sudo chmod 600 /etc/samba/credentials
```

Теперь добавьте в **/etc/fstab** такую строку:

```
//server/share /mount/point cifs credentials=/etc/samba/credentials 0 0
```

Если для доступа к разным машинам у Вас разные пароли, создайте файл **credentials** для каждой из них, например, **/etc/samba/credentials.имя_хоста**. Кроме того, общий доступ к файлам Linux и других Unix-подобных систем, а также Mac OS, возможен через NFS (Network File System). На подробности здесь места нет, но все необходимое найдется в **LXF117**. НБ



› Чтобы видеть разделяемые сетевые каталоги Windows из среды Linux, их надо смонтировать.

3 Монитор на час

Я настроил *Monit* для тестирования соединения с сервером FreeNAS. Но сервер подключается только с 7:00 до 17:00. Как настроить *Monit*, чтобы он в период простоя не отправлял мне никаких извещений о проблемах, но был активен в часы работы сервера? Механизм тайм-аута мне вряд ли поможет, потому что днем уведомления все-таки нужны.

У меня такая конфигурация:

```
Check host FREENAS with address 192.168.3.24
if failed icmp type echo then alert
if failed port 22 proto ssh then alert
```

bobthebob1234, вопрос с форумов

Monit поддерживает активизацию и блокировку проверки сервисов, как индивидуально, так и группами. Он не будет автоматически вести за сервисом наблюдение, если добавить к файлу конфигурации такую строку:

```
mode manual
```

Затем воспользуйтесь *Cron* для запуска следующей команды в 7:00 или чуть позже этого момента, чтобы дать время серверу FreeNAS стартовать:

```
monit monitor FREENAS
```

Перед 17:00 выполните команду еще раз, но уже с опцией **unmonitor**. Кроме того, это можно проделать и для группы сервисов, добавив в раздел каждого из них в **monitrc** следующую строку:

```
group имя_группы
```

С помощью *Cron* запустите следующую команду:

```
monit -g имя_группы monitor
```

Почтовые сообщения по-прежнему будут генерироваться, потому что по умолчанию *Monit* отправляет уведомления обо всех событиях, включая административные действия. Это поведение можно изменить, добавив в **monitrc** строку

```
alert foo@bar but not on {instance}
```

Теперь извещения будут отправляться в ответ на все события, кроме тех, что связаны с запуском или остановкой сервиса [instance events]. Можно также попробовать вариант

```
alert foo@bar only on {timeout, noexist}
```

Извещения будут отправляться только тогда, когда наступят эти два конкретных события. Подробная информация об извещениях и всем необходимом найдется на man-странице программы *Monit*.

Мы не учли еще один аспект. *Monit* запоминает состояние каждого сервиса между остановками и перезагрузками, и если перезапустить *Monit* или сервер, на котором он выполняется, непосредственно перед моментом одного из изменений, он будет работать, когда не должен, и наоборот. Можно просто настроить *Cron* на ежечасную отправку команды **monitor** или **unmonitor**, или написать скрипт, чтобы он срабатывал при старте и проверял время, а затем отправлял соответствующую команду. Вам должен помочь простой сценарий примерно следующего вида:

```
HOUR=$(date + %H)
If [[$HOUR -ge 7]]&&[[$HOUR -lt 17]]; then
monit monitor FREENAS
else
monit unmonitor FREENAS
fi
```

Удачи! НБ

4 Печать и браузеры

Я работаю в Windows еще со времен Windows 3, но после появления первых версий Knoppix Live CD начал эксперименты и с Linux. И практически с каждым опробованным дистрибутивом у меня возникали проблемы.

Печать: разве разработчики Linux никогда ничего не печатают? Например, Ubuntu 9.04 даже не распознает мой принтер!

Я постоянно натыкаюсь на Web-сайты, нормально открывающиеся только в *Internet Explorer*. Да-да, именно так, и *Firefox* не спасает. Как это обойти? По-моему, *IE6* можно использовать и в Linux, но я не знаю, как это делать.

Существует ли дистрибутив Fedora на Live CD? Я не хочу делить мой жесткий диск на разделы, и вдобавок все инструменты, которые я перепробовал, в том числе *Acronis*, *Easeus* и *GParted*, сообщают, что так нельзя, потому что диск «неправильно отформатирован». Я этого не понимаю.

Джон Голдспинк [John Goldspink]

К сожалению, Вы не указали модель Вашего принтера. Иногда случается так, что CUPS, система печати Linux, неправильно идентифицирует даже поддерживаемый принтер и в результате загружает не тот драйвер. Лучшее, что Вы можете сделать – загляните в базу данных фонда Linux Foundation, известную как Linux Printing Database, на сайте www.linuxfoundation.org/collaborate/workgroups/openprinting/database/databaseintro. Поищите там свою модель принтера, и Вы узнаете, какой драйвер лучше всего для него подходит и как его установить и настроить в CUPS. Помимо прочего, правильный драйвер поможет повысить и производительность печати.

Ну, а сайты, работающие только с *Internet Explorer* – на совести тех, кто их разрабатывает и поддерживает. Web диктует четко определенные стандарты, и сайт, способный работать только с одним браузером, эти стандарты нарушает. Нередко такие сайты и не нуждаются в *IE*, просто их разработчики не удосужились проверить их работу с чем-то другим: обычно они проверяют идентификатор пользовательского агента [User-Agent ID], отправляемый с каждым запросом, и от- »



› Если без *IE* не обойтись, запустите его в Linux (правда, не последнюю версию) через *Wine*.

клоняют запрос, если это не *IE*. Тогда требуется лишь настроить Ваш браузер так, чтобы он «представлялся» сайту как *Internet Explorer*: зачастую это срабатывает.

В *Konqueror* есть встроенное средство для этой цели. В *Firefox* или *Chromium* потребуется установить расширения, *User-Agent Switcher* и *Chromeleon User-Agent Spoofer*, соответственно.

Если же доступ к сайту действительно дает только *IE*, можно запустить его через *Wine*.

Wine эмулирует библиотеки и системные сервисы Windows, чтобы Windows-приложения могли работать под Linux. Считается, что установить и сконфигурировать *Wine* очень сложно, но, к счастью, пакет *Ies4linux* упрощает эту задачу для *IE*. Его можно добыть по адресу www.tatanka.com.br/ies4linux или через менеджер пакетов Вашего дистрибутива. Он автоматически выполнит шаги, требуемые для запуска *Internet Explorer* в Linux.

Установите *Ies4linux*, запустите его в окне терминала и следуйте инструкциям – они подскажут, как загрузить и развернуть различные версии *IE*. Но для этого надо иметь лицензию на Windows. Предоставляется только *IE6* (хотя в последней версии есть кнопка Advanced, нажав которую, можно получить бета-версию *IE7*). Впрочем, применяя подмену User Agent, а также посылая законные жалобы владельцам web-сайтов, Вы добьетесь доступа к большинству из них.

Теперь – на последний Ваш вопрос. Проект Fedora предоставляет установочный LiveCD, и можно сначала попробовать дистрибутив, а затем, если он понравится, инсталлировать его с того же диска. Это первая опция на странице, откуда предлагается скачать Fedora 12 (http://fedoraproject.org/en/get-fedora). Скорее всего, ошибка с разбиением диска на разделы связана с тем, что на Вашем винчестере имеется скрытый раздел восстановления Windows: на новых компьютерах он заменяет дистрибутивный компакт-диск с копией Windows. **НБ**

4 Проблемные разделы

В Я наконец рискнул удалить Windows, выбрав Ubuntu 9.10. Затем я поумнел и установил вместо нее OpenSUSE 11.2. Но она меня не совсем устроила, и я решил вернуться к Ubuntu и отвел под нее весь жесткий диск. Все шло хорошо почти до конца, но тут все обернулось так, как будто я работаю с Live CD, а компьютер сообщил, что не находит *Grub*.

После этого я снова вернулся к Windows и попытался создать систему с двойной загрузкой, но безуспешно. Проверка через дисковый менеджер в Windows показала, что раздел Linux пуст.

Майкл Лукани [Michael Lucani]

О Вы хотите установить Ubuntu как систему с двойной загрузкой или как единственную ОС? Для создания системы с двойной загрузкой Вам необходимо при установке сообщить инсталлятору, что он должен использовать имеющийся пустой раздел Linux.

Пустые разделы Linux могут запутать установщик, потому что если на разделе есть файловая система, то эта часть диска не пуста по-настоящему, даже если файлов на ней никаких нет. Часто простейшим способом решения этой проблемы будет загрузиться с любого LiveCD, воспользоваться менеджером разделов (для Ubuntu это будет выглядеть так: Система > Администрирование > GParted [System > Administration > GParted]) и удалить пустой раздел. Сохраните внесенные изменения, перезагрузитесь, чтобы система смогла увидеть новую раскладку диска, а потом запустите установщик. Теперь он обнаружит, что часть диска занята копией Windows, а часть является невыделенным пространством. По умолчанию инсталлятор предложит установить Ubuntu на свободное место и сконфигурирует загрузочное меню так, чтобы Вы могли выбирать, какую ОС запустить.

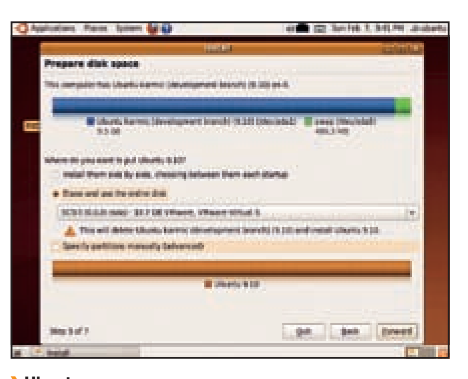
Если Вам хочется установить Linux как единственную ОС, то, дойдя до момента, когда установщик отобразит экран Подготовка диска [Prepare Disk Space], выберите опцию Использовать весь диск [Erase And Use Entire Disk]. После этого убедитесь в том, что в главную загрузочную запись (Master Boot Record, MBR) будет записан загрузчик *Grub*, на следующем экране установщика. Если этого не сделать, то жесткий диск не станет загрузочным, и Ваш компьютер, скорее всего, продолжит попытки загружаться с CD, снова и снова возвращая Вас в среду LiveCD.

6 Пропал жесткий диск

В У меня Mandriva 2009, и я пользуюсь внешним жестким диском с интерфейсом USB для резервного копирования. Когда этот диск не нужен, он отключается. Все работало хорошо — когда я подключал диск и выбирал опцию Открыть в новом окне [Open In New Window], файлы и папки отображались в *Konqueror*. Я ничего не менял (не считая возможных обновлений), но теперь, подключив диск, вижу только сообщение 'Permissions Denied' — даже при входе в систему как root. Диск USB не отображается на странице Storage Media [Накопители] — виден только основной жесткий диск. В каталоге /dev есть значки для разделов sda, sda1, sda2, sda5, sda6, sdb, sdc, sdd, sde, sdf и sdf1. С виду похоже, что sda1 и sda6 — это основной жесткий диск, и у них одинаковые права доступа — пользователь root и группа disk. Буду очень благодарен за любые подсказки, как вернуть доступ к моему жесткому диску USB.

Дэйв Притчард [Dave Pritchard]

О Судя по описанной Вами ситуации, Вы, скорее всего, отключили свой диск USB, предварительно не отмонтировав его. Как правило, при щелчке правой кнопкой мыши по значку USB-диска на экране должно появляться меню с опцией 'Safely Remove' [Извлечь] или 'Unmount Volume' [Отмонтировать]. Эту операцию необходимо проделывать при всяком отключении какого бы то ни было съемного записываемого устройства, будь то жесткий диск или USB-брелок. Не сделав этого, Вы рискуете оставить файловую систему во внутренне противоречивом состоянии или даже серьезно повредить ее, в зависимости от того, что происходило в момент отключения накопителя. Для повышения производительности работы, операционные системы буферизуют данные в памяти и записывают их на диск с небольшим запаздыванием. Если отключить накопитель в этот краткий период, неприятности гарантированы.



› Ubuntu дает возможность стереть содержимое жесткого диска перед установкой.

Впрочем, хорошая для Вас новость такова, что обычно подобные повреждения можно устранить, обработав каждую из затронутых файловых систем командой *fsck*. Судя по Вашему списку **/dev**, интересующее нас устройство – **/dev/sdf** с файловой системой на **/dev/sdf1**. Поэтому попробуйте запустить в терминале следующую команду (от имени root или с префиксом **sudo**):

```
fsck /dev/sdf1
```

Не пытайтесь сразу же примонтировать диск, а запустите команду *fsck* и дождитесь завершения ее работы, что требует времени. Возможно, в процессе работы программа задаст Вам некоторые вопросы, в зависимости от типа проверяемой файловой системы.

Когда *fsck* завершит работу, снова примонтируйте диск и загляните в каталог **lost+found** (если раздел был отформатирован под файловую систему ext3). Туда *fsck* поместит все файлы, которые не удалось идентифицировать. Вам останется уже самим определять, что это за файлы и куда их переместить.

Эту работу также можно выполнить, загрузившись с LiveCD; а если поврежден Ваш системный диск, без LiveCD и не обойтись. **НБ**

7 YDL6 на PS3

В Я успешно установил Yellow Dog Linux 6.1 на мою игровую приставку PlayStation 3, поделив на разделы добавленный жесткий диск SATA на 500 ГБ. По части Linux я, конечно, полный профан, но все же решил попробовать, потому что мне нравится идея иметь на PS3 настоящую ОС. Затем я попытался установить беспроводное сетевое соединение, но безуспешно. Все мои попытки завершались с сообщением 'determining IP information for wlan0... failed'. Не прольете ли вы свет на эту загадку? Не уверен, что YDL 6.1 — лучший выбор для Wi-Fi, но установил именно его, потому что хотел использовать и все прочие компоненты PlayStation 3 (а YDL вроде бы подходит для этой цели лучше других дистрибутивов). Поэтому другой мой вопрос будет таким: какой дистрибутив Linux вы бы рекомендовали для PlayStation 3?

Томас Холден [Thomas Holden]

О Причин у Вашей проблемы может быть несколько, но по сути PS3 «не сговорилась» с точкой доступа к беспроводной сети (Wireless Access Point, WAP). Вы не сообщили тип точки доступа и Вашего маршрутизатора, но в любом случае первым делом при поиске неполадок в работе беспроводных сетей нужно отключить шифрование на точке доступа. Хотя для повседневной работы это очень плохое решение, в Ваших обстоятельствах оно может быть полезным, потому что устраняет одну из возможных, и достаточно частых, причин ошибки (предоставленные вами сведения позволяют предположить, что вы используете схему доступа без аутентификации – Open – и неверно указываете пароль для кодирования трафика). Добившись работы соединения без шифрования, его можно будет снова включить и устранять неполадки по одной за раз.

Если и после отключения шифрования соединение не заработало, проверьте журналы на маршрутизаторе. Обычно это делается через его web-интерфейс, но может потребоваться компьютер, подключенный к маршрутизатору Ethernet-кабелем (некоторые маршрутизаторы по соображениям безопасности не допускают администрирования через беспроводные сети). Не установлена ли на маршрутизаторе какая-либо блокировка MAC-адресов? Получает ли он запросы на установление соединения от PS3? Можно также запустить на приставке сеанс терминала и набрать

```
iwlist wlan0 scan
```

Эта команда проверяет, обнаруживает ли PS3 Вашу точку доступа. Только сперва убедитесь, что маршрутизатор не настроен так, чтобы скрывать свой SSID. Кроме того, попробуйте команду

```
iwconfig wlan0
```

Она покажет, подключены ли Вы к точке доступа, причем к той, которая Вам нужна. Я сам один раз долго мучился, пока не осознал, что пытаюсь подключиться не к своей точке доступа, а к маршрутизатору соседа.

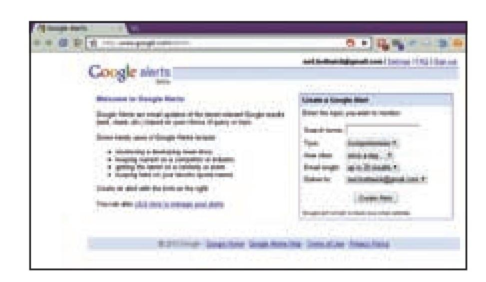
Yellow Dog – вполне приличный дистрибутив, но не самый популярный среди пользователей Linux. Есть вариант Ubuntu, предназначенный специально для PS3 – PSUbuntu (http://psubuntu.com). Он имеет собственные форумы и wiki-страницу; но и на форумах и сайте Ubuntu Вы тоже обнаружите достаточное количество информации.

Ваша проблема не должна быть аппаратной – присутствие **wlan0** показывает, что оборудование, как минимум, поддерживается и распознается. Скорее всего речь идет о дефекте настройки, и в Ubuntu проблема может и не появиться. **НБ**

8 Персональные пауки

В Мои служебные обязанности включают отслеживание замечаний про компанию и ее продукты, публикуемых в Web. Я знаю, что для этого можно применить Google и другие поисковики, но я задумался: нельзя ли автоматизировать этот процесс? Нет ли программы для Linux, способной принимать список ключевых слов и посылать мне результаты? А лучше всего, если бы она запускалась через *Cron*.
Джефф Слэк [Geoff Slack]

О Для этого Вам даже не нужна отдельная программа. Google предлагает сервис извещений [alerts], выполняющий именно то, о чем Вы спросили. Пройдите по ссылке www.google.com/alerts и введите свою искомую строку – лучше будет сперва выполнить ручной поиск, а затем настроить и уточнить его для получения должных результатов. Затем укажите адрес электронной почты, куда отправлять сообщения, максимальное количество результатов на одно сообщение, а также периодичность их получения: еженедельно, ежедневно или по мере нахождения совпадений [as it happens]. В последнем случае почтовое извещение будет отправляться Вам всякий раз, когда «пауки» Google обнаружат вхождение указанной Вами строки. При желании можно указать в настройках, чтобы результаты отправлялись в виде RSS-канала на Google Reader, а не по электронной почте.



› Google Alerts может оповещать вас о каждом результате для вашего поискового запроса по электронной почте или через RSS-канал.

Эти настройки не глобальные: Вы можете настраивать их отдельно для каждого своего поиска. Получаемые почтовые сообщения будут содержать примерно ту же информацию, которую Вы бы увидели на обычной странице результатов поиска, с тем исключением, что каждое новое сообщение доставит результаты, которые Вы еще не видели. **ПХ**

9 Право печатать

В Пару лет назад мне подарили на день рождения замечательный принтер HP LaserJet 1020. Недавно у меня появилось желание полностью перейти на Linux, и единственное, что удерживает меня в Windows — это он. Сейчас я использую Linux Mint Gloria на своем настольном компьютере и Linux Mint Helena на ноутбуке. Принтер отказывается общаться с обеими «дамами». »

Часто задаваемые вопросы

Брандмауэры

» Что такое брандмауэр, и с чего мне бояться пожара на моем компьютере?
Брандмауэр [firewall] – это комбинация аппаратных и программных средств, защищающая компьютер(ы) от несанкционированного доступа. Самое распространенное его применение – защита компьютера или сети от недостойных обитателей Интернета.

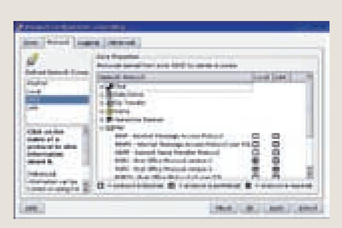
» Так брандмауэр — это программа или отдельный «ящик»?
Бывает и так, и этак. Аппаратные брандмауэры обеспечивают лучшую защиту, поскольку они останавливают злоумышленника еще до того, как он добрался до компьютера. Встроенным брандмауэром снабжены многие широкополосные маршрутизаторы, так что налетчиков остановят уже у модема. Еще один вариант – выделить под брандмауэр старый компьютер со специализированным дистрибутивом, типа IPCop.

» У меня только один компьютер; могу ли я запустить брандмауэр на нем?
Да, можете, и очень эффективный. Конечно, не так, как аппаратный, но злодеев отгонит. Истинные параноики ставят оба.

» А какие есть варианты?
Функции брандмауэра в Linux возложены на утилиту *Iptables*, пользующуюся компонентом ядра под названием *Netfilter*. Настраивается *Iptables* заданием серии правил, определяющих, что разрешено, а что – нет. Написание правил для *Iptables* – штука сложная; допустив ошибку, вы получите брандмауэр, выполняющий не то, что надо. Различные программы для брандмауэров, от основанной на скриптах *Shorewall* (www.schorewall.net) до графических приложений вроде *Guarddog* (http://bit.ly/fEkz8) и *FirewallBuilder* (www.fwbuilder.org), упрощают процесс создания правил. Все они генерируют скрипты для *Iptables*, и с любой из них вы создадите правила, пригодные для сервера, работающего без графической системы *X Window*.

» Но ведь Linux безопасен, разве нет? Зачем мне еще и это?
Да, но только если вы сами об этом позаботитесь. Замки на дверях не остановят грабителя, если окна у вас настежь. Это касается и безопасности в Интернет. Linux предусматривает средства, способные защитить вашу сеть от вторжений, но их надо еще и применить! Конечно, наличие прав доступа не даст злоумышленникам особо развернуться, но уж лучше вовсе не пускать их в свою систему.

» Да у меня и украсть-то нечего!
Вам это только кажется. Например, в кэше вашего браузера хранятся страницы финансовых сайтов, которые вы посещаете. Пусть там и нет паролей, но все равно это ценная подсказка для вора. Ваши почтовые сообщения содержат личную информацию. Наконец, ваше интернет-соединение могут употребить для рассылки спама, и ваш провайдер накажет вас – вплоть до заблокирования вашей учетной записи. Причем ни одно из этих злодеяний не требует прав root: достаточно быть обычным пользователем.



› В программе *Guarddog* брандмауэр настраивается через графический интерфейс.

Продолжительный поиск рецептов для решения моей проблемы привел только к выводу, что мне «повезло», и с аналогичными моделями (10x) у большинства людей возникают такие же трудности. Найденные рецепты помогали, но до первой перезагрузки: независимо от того, загружалась ли я потом в Windows или в Mint, принтер отказывался печатать. Судя по крутящимся валикам, прошивка не месте, однако в диспетчере устройство определяется как недоступное или занятое.

Очень надеюсь, что Вы поможете мне в нелегкой борьбе за свободу печати.

Марина Капицына

О На сайте http://www.openprinting.org/printer/HP/HP-LaserJet_1020 сообщается, что Ваш принтер «в основном» работает. Там же рекомендуются драйверы *foo2zjs*, а не *HPLIP*, которые, судя по всему, выбрали Вы. Для работы данного принтера необходимо загрузить на него прошивку (firmware, arm binary) – «либо с помощью скрипта для *Hotplug/udev*, идущего в составе *foo2zjs*, либо вручную». При обнаружении принтера ОС должна сделать это автоматически, и затем работать с LJ 1020 как с обычным устройством.

В системах на базе Ubuntu пакет *foo2zjs* установлен по умолчанию. Прошивки же нужно скачивать отдельно с сайта foo2zjs.rkkda.com. В состав пакета входят нужные утилиты: *getweb* (загрузка прошивки) и *arm2hpdl* (добавляет необходимые для принтера HP заголовки к двоичному файлу arm, см. **man arm2hpdl**). После добавления заголовков прошивку можно «заливать» в принтер (этим и занимается *udev/hal*). Чтобы не проделывать всё вручную, можно применить следующий скрипт:

```
#!/bin/bash
printers="1000 1005 1018 1020"
cd /tmp
for prn in $printers; do
  img="sihp${prn}.img"
  dl="sihp${prn}.dl"
  getweb $prn
  if [ -f $img ]; then
    arm2hpdl $img > $dl
    sudo cp $dl /usr/share/foo2zjs/firmware/$dl
    rm $img $dl
  fi;
done
```

Скопируйте его в ваш домашний каталог **~/bin**, сохраните, скажем, с именем **HP1020.sh** и сделайте исполняемым с помощью

```
cd ~/bin
sudo chmod +x HP1020.sh
```

Затем запустите его

```
sh HP1020.sh
```

и наберите в консоли

```
cat /usr/share/foo2zjs/firmware/sihp1020.dl > /dev/usb/lp0
```

где **/dev/usb/lp0** – файл устройства принтера. Имя может быть другим, например **/dev/usblp0**. Чтобы узнать наверняка, наберите

```
dmesg | tail -20
```

после включения принтера. Теперь он должен заработать. **ЮД**

10 Загрузка с ISO

В статье про *Grub 2* в LXF128 говорится, что он умеет загружать ОС из ISO-образа. Меня это очень заинтересовало: существует много LiveCD/DVD, которые хотелось бы использовать, но не очень удобно устанавливать. Я уж не говорю о возможности сборки собственного респина под конкретные задачи и распространения его на большое количество машин и о прочем. Применений масса, но вот примера такой конфигурации *Grub* с комментариями в статье не было. Как она выглядит?

Алексей Маслий

О Самый простой способ достичь желаемого эффекта – воспользоваться консолью загрузчика (обычно она вызывается горячей клавишей ‘c’):

```
sh:grub> loopback loop (hd0,2)/downloads/ubuntu-9.10-desktop-amd64.iso
sh:grub> linux (loop)/casper/vmlinuz iso-scan/filename=/downloads/ubuntu-9.10-desktop-amd64.iso root=/dev/sda2 boot=casper noeject noprompt
sh:grub> initrd (loop)/casper/initrd.lz
sh:grub> boot
```

В приведенном примере мы запускаем Ubuntu 9.10 Desktop LiveCD (amd64) из файла **ubuntu-9.10-desktop-amd64.iso**, расположенного в каталоге **downloads** на файловой системе ext4 второго первичного раздела первого жесткого диска. Разумеется, названия дисков, разделов, а также имена файлов и пути доступа к ним в вашем случае могут быть другими.

Важно понимать, что по сравнению с предыдущими версиями, Grub 2 только упрощает настройку, а поддержка загрузки с образа ISO должна присутствовать в дистрибутиве (в нашем случае, за это отвечает опция **iso-scan/filename**). И, конечно, если вы используете другой образ, передаваемые ядру параметры будут отличаться. Обо всем этом можно прочесть на форумах конкретного проекта.

Если вы не хотите работать в консоли *Grub 2*, загрузку можно прописать в меню (**/boot/grub/grub.cfg**):

```
menuentry "Ubuntu 9.10 Desktop AMD64 (ISO)" {
   loopback loop (hd0,2)/downloads/ubuntu-9.10-desktop-amd64.iso
   linux (loop)/casper/vmlinuz iso-scan/filename=/downloads/ubuntu-9.10-desktop-amd64.iso root=/dev/sda2 boot=casper noeject noprompt
   initrd (loop)/casper/initrd.lz
}
```

Как говорилось в статье, редактировать его напрямую не стоит. Нужно менять шаблоны (обычно они расположены в каталоге **/etc/grub.d**) и файл **/etc/default/grub**, после чего перегенерировать конфигурацию загрузчика. **ЕК**



Коротко про...

Компиляцию ядра

Случалось ли вам видеть такой ответ на вопрос: «Перекомпилируйте свое ядро с опцией X»? Хотя он и звучит пугающе, но это довольно простая задача – изменить существующую настройку вашего ядра. Конфигурирование ядра «с нуля» чуть сложнее.

Во-первых, убедитесь, что у вас установлен правильный пакет исходных текстов ядра – его версия должна соответствовать работающему у вас ядру. Далее, понадобится компилятор *GCC* и его эскорт; большинство дистрибутивов имеют в своем составе пакет типа **build-essentials**, содержащий все необходимое для компиляции ПО.

Запустите терминал от имени root (в Ubuntu можно набрать команду **sudo bash**) и введите такие команды:

```
cd usr/src/linux
make menuconfig
```

Разверните окно терминала, чтобы видеть все предлагаемые настройки. Пакет с исходным кодом ядра должен включать файл, содержащий опции, использованные для сборки вашего текущего ядра; вам требуется только найти, что надо менять.

Если это не так, отыщите требуемый файл в каталоге **/boot**. Скопируйте его в **/usr/src/linux/.config**, проделайте те же шаги и нажмите клавишу **/**, чтобы вызвать строку поиска, и введите имя искомого элемента. В результате вы получите путь к нему. Закройте результаты поиска и воспользуйтесь клавишами курсора и Enter, чтобы добраться до нужного вам элемента. Выберите его нажатием клавиши Пробел. Повторите эти действия для всех опций, которые требуется изменить, затем выйдите с сохранением.

Теперь скомпилируйте и установите ядро командой

```
make all modules_install install
```

Она соберет все, что нужно, установит модули ядра, а затем копирует само ядро в каталог **/boot**. Кроме того, команда устанавливает символические ссылки [symlinks] на текущее ядро из **vmlinuz** и предыдущее – из **vmlinuz.old**. Поэтому, если в вашем меню *Grub* есть опции для загрузки двух ядер, менять ничего не потребуется, и всегда возможен откат к старой версии ядра.

Теперь перезагрузитесь – и поздравьте себя с первой перекомпиляцией ядра. В случае проблем, загрузите старое ядро и еще раз посмотрите опции, которые вы меняли.

Большой вопрос Как сделать, чтобы Network Manager не спрашивал у меня пароль?

11 Беды с паролями

В Я пользуюсь Ubuntu, и у меня есть пара проблем с *Network Manager*. При каждом запуске он требует мой пароль — нельзя ли это пресечь? Без ввода пароля не активируется даже беспроводное соединение. И если я включаю компьютер, чтобы наскоро проверить почту, это делается отнюдь не быстро. Как избавиться от диалогового окна, предлагающего ввести пароль, и ускорить процесс выхода в сеть?
Кевин Болдуин [Kevin Baldwin].

О *Network Manager* работает от имени обычного пользователя, а ключи шифрования для беспроводных соединений хранятся на «связке ключей» пользователя. *Network Manager* может получить их после Вашей регистрации в системе и ввода пароля. При автоматической регистрации [auto-login] метод не сработает. Однако для *Network Manager* существуют более простые и быстрые альтернативы.

Wicd (http://wicd/sourceforge.net) – программа-менеджер проводных и беспроводных соединений. Она пока не работает с 3G-модемами, а только с Ethernet и Wi-Fi. Подход *Wicd* принципиально отличен от *Network Manager*: это сервис, стартующий при запуске. Для его подключения не нужно, чтобы был запущен рабочий стол или даже чтобы был зарегистрирован пользователь.

К моменту загрузки рабочего стола беспроводное соединение обычно уже установлено и аутентифицировано. Применять глобальную настройку во многопользовательской среде – не идеальный вариант, но для ноутбуков и нетбуков, обычно имеющих одного владельца, это ускоряет работу. В составе программы есть еще настольное клиентское приложение для наблюдения за сетями и переключения между ними. *Wicd* использует стандартные сетевые инструменты для сервера и *GTK* – для клиента, так что о зависимостях говорить не приходится, и хорошо работает в любой настольной среде.

Установив *Wicd*, убедитесь, что сервис работает, раскройте окно клиента, перейдите в группу опций Preferences [Настройки] и задайте устройства для ваших беспроводных и проводных соединений. Изменив какое-либо из этих устройств, нажмите кнопку Refresh [Обновить], чтобы программа просканировала доступные беспроводные сети. Для задания настроек безопасности своей беспроводной сети нажмите кнопку Properties [Свойства] – для прочих настроек обычно сойдут значения по умолчанию. Программа запросит пароль – или root, или пользователя, в зависимости от того, использует ли ваш дистрибутив **su** или **sudo**. Сервисы запускаются глобально, и настройки хранятся в файле конфигурации – его чтение и запись требуют прав root.

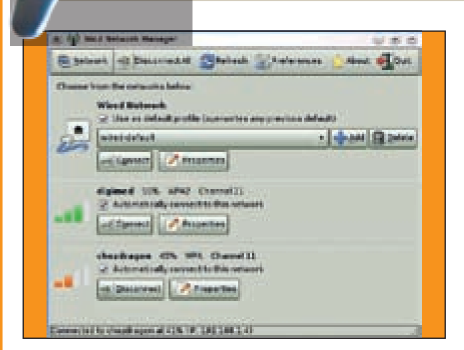
Wicd поддерживает шифрование WEP, WPA и WPA2 и определяет тип шифрования, используемого в конкретной сети. Эти данные хранится в файле **/etc/wicd/wireless-settings.conf**, и *Wicd* при запуске предоставляет к нему доступ с правом чтения только пользователю root.

Каждое соединение можно настроить отдельно: скажем, настроить почтовый клиент на автоматический переход в оффлайн-режим при переходе в оффлайн любого из интерфейсов, а сетевые ресурсы монтировать только при включении домашнего или рабочего интерфейса. Например, вот скрипт, переводящий мою почтовую программу *Claws* в автономный режим:

```
claws-mail --offline
```

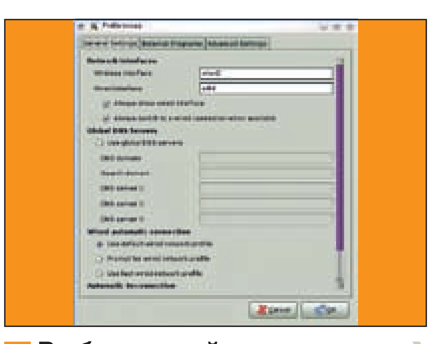
Аналогичный скрипт у меня переключает программы в режим онлайн. **НБ**

Шаг за шагом: Управляем сетью через Wicd



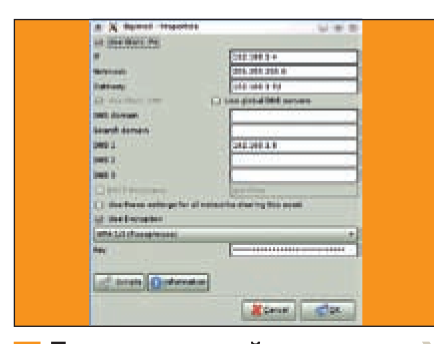
1 Запуск клиента Wicd

Настольный клиент *GTK* управляет всеми задачами по настройке *Wicd*, включая выбор сетей для автоматического подключения



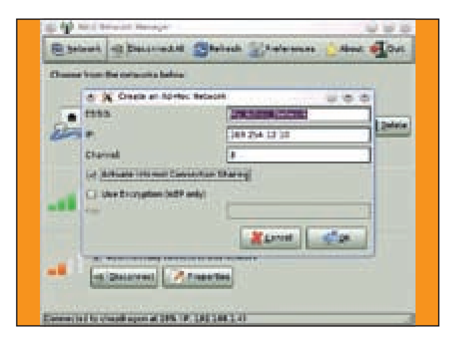
2 Выбор устройств

Перейдите в раздел Preferences и убедитесь в том, что для всех ваших проводных и беспроводных сетей выбраны правильные сетевые интерфейсы.



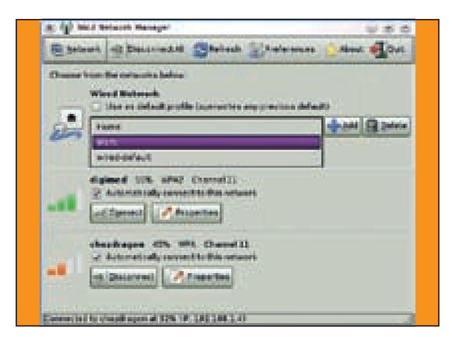
3 Личные настройки

Каждая сеть может использовать глобальные настройки или иметь собственную сетевую конфигурацию. Помимо прочих настроек, здесь задаются и ключи шифрования.



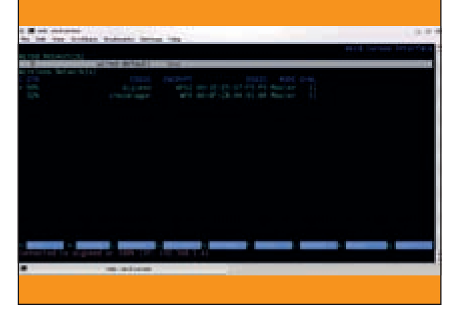
4 Сети Ad-Hoc

Wicd устанавливает прямые соединения между компьютерами без центральной точки доступа и предоставляет совместный доступ к интернет-соединению.



5 Проводные сети

Проводные сети тоже поддерживаются, и вы можете создавать профили для различных местоположений, каждый со своими настройками

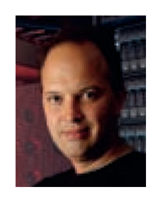


6 Подключение без GUI

Клиент *wicd-curses* может запускаться из командной строки и управлять работой *Wicd* без всякого рабочего стола. LXF



LXFHotPicks



Ник Вейч
Ник компилирует Hotpicks, используя текстовый редактор и головной мозг. А иногда — только текстовый редактор.

QMotion » Fubar » PCI Utilities » Electric » Gnumeric » XSensors » Worker » Turious » Kajongg » Jailkit » Florence

---

Утилита для web-камеры

QMotion

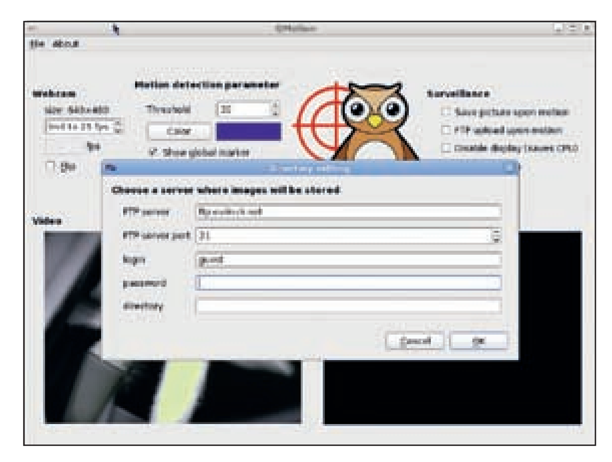
Версия 2.1 **Сайт** http://slist.lilotux.net/linux/qmotion/index_en.html

Пару раз сумбурно пообщавшись через web-камеру с родственниками, живущими на другой стороне планеты, вы вскоре зададитесь вопросом: нельзя ли ее приспособить к чему-то действительно полезному?

В первую очередь напрашивается видеонаблюдение, но поскольку вряд ли кто захочет заполнять свой жесткий диск часами изображения пустой комнаты, появились программы-детекторы движения.

Для превращения web-камеры в инструмент слежения имеется несколько сходных проектов, но *QMotion* стоит внимания хотя бы за крутой интерфейс, показывающий ту зону изображения, где что-то движется, весьма забавным и не лишенным привлекательности способом.

Зоны, где происходит нечто интересное, обозначаются специальным маркером, и вы можете наблюдать плавно исчезающие различия между кадрами в специальном поле. Алгоритмические хитрости предоставлены библиотекой *OpenCV*. Однако, в отличие от более серьезных проектов-конкурентов, здесь нельзя ограничиться движением только в некой части поля зрения камеры.



› *QMotion* умеет автоматически отправлять снимки при движении чего-либо, то есть можно дистанционно наблюдать за вашим столом.

> «Он показывает ту зону изображения, где что-то движется.»

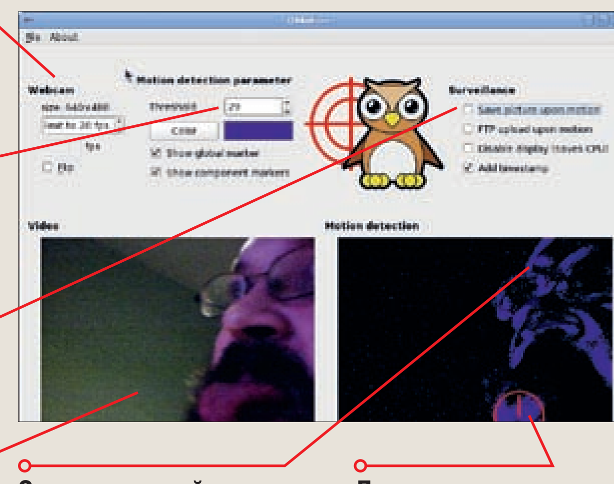
Слепая зона

Поясним на примере: возможно, вам надо сохранять все изображение, но только если кто-то вошел в определенную дверь – а этого сделать нельзя. К сожалению, это ограничивает пользу *QMotion* в качестве проекта для круглосуточного обеспечения безопасности дома или офиса, но с ним определенно интересно работать, наблюдая за бассейном с рыбками.

Последовательность медленно сходящих на нет изображений области изменений вызывает в памяти живое рекламное видеомикширование *Top Of The Pops* конца 1970-х, и стоит потратить пять минут, чтобы увидеть, сколько движений, обведенных кругом, вы создадите на экране, просто мотая головой. Возможно, мы не слишком требовательны, но среди удовольствий, доставляемых web-камерой, это заняло второе место.

Исходный код достаточно прост, но требует инструментария разработчика и для *Qt 4*, и для KDE 4.x, а также вышеупомянутой библиотеки *OpenCV*. Исходные тексты организованы как проект *Qt*, так что используйте **qmake** перед обычной процедурой **make** и **make install**.

Исследуем интерфейс QMotion



Настройка оборудования
Предоставляет информацию об имеющейся web-камере и параметрах скорости.

Порог
Задайте порог чувствительности для более тонкого детектирования движения.

Средства наблюдения
Сохраняйте или выгружайте изображения, когда *QMotion* определит движение.

Живое отображение
Показывает живую картинку с опциональной отметкой времени.

Зоны изменений
Зоны, где произошли изменения, на этом экране окрашены в пурпурный цвет.

Датчик движения
Все, что шевелится, выделяется кружком с указанием направления движения внутри.

Работа с изображениями

Fubar

Версия 1.0 **Сайт** www.4cko.be/fubar-www

Из ничего – ничего и не сделаешь, но порой, при попутном ветре, можно взять что-нибудь, малость распушить, и получится нечто годное хотя бы для латания дыр. С рыбой и хлебами сработало, но эта программа пытается внести в изображение деталь, которой там на самом деле просто нет. Любой владелец цифровой камеры знает, что, дойдя до уровня пикселей, подробностей больше не извлечь.

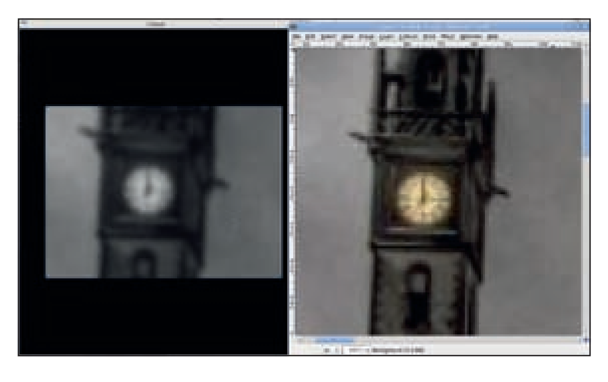
Увы, реальная жизнь – это вам не голливудский боевик, где можно все больше и больше увеличивать изображение конверта в верхнем кармане человека, бредущего по улице, пока не разглядишь на нем почтовый индекс. Или все-таки?..

Еще раз увы. И тем не менее *Fubar* – интересная штука. Он применяет к изображениям технологию под названием «кригинг» (krigging). Это нечто среднее между обработкой сигналов и основанным на догадке восстановлением недостающих элементов, с применением вейвлет-методик. Следует отметить, что особого успеха тут не добьешься, даже если напрячь воображение. Управляйте усечением и вариацией функции с помощью клавиатуры (когда вы запустите программу, в ней будет указано, как это делается), и получите этакие суповые разводы.

Кристально ясно?

Нажав на пробел, вы примете текущие установки за новый источник, так что можно увеличить нужный вам эффект. Конечно, никаких номерных знаков, видных при 12 пикселях, но своего рода головокружительное путешествие вам гарантировано. *Fubar* поддерживает только изображения PPM, и если вы хотите загрузить собственные, придется заготовить их конвертировать – и здесь вам помогут пакет **netpbm** и компания.

> «Программа пытается внести в изображение деталь, которой нет.»



› Изображение справа — оригинал; а слева — то, что вышло после обработки в *Fubar*.

Если вы распакуете исходники и введете **make**, то есть шанс, что они не соберутся. Чтобы все заработало, нужны заголовочные файлы для *Glut*. Сперва установите пакет **freeglut-devel** своего дистрибутива. После этого сборка пойдет легко и просто. Скомпилировав программу, запустите ее командой

```
fubar in.ppm
```

Загрузится тестовое изображение. Можно увеличивать картинку левой кнопкой мыши и уменьшать правой – объяснения приводятся на экране-заставке. Для желающих более подробно узнать о вейвлет-теории, на которой основана работа данного приложения, на сайте представлен весьма полезный PDF.

Системный инструмент

PCI Utilities

Версия 3.1.7 **Сайт** http://mj.ucw.cz/pciutils.html

Linux – вещь поразительная. Мы имеем в виду не открытость кода или свободу, предоставляемую данной платформой, и тому подобное: поразительно то, что он вообще работает.

ОС – это ведь не только ядро: есть бессчетное количество критически важных системных программ (честно говоря, по большей части это – инструменты GNU), которые позволяют этому мультиплатформенному монстру-тяжеловесу взлететь.

И все эти столпы приложений как бы скрепляет раствором код, обеспечивающий их совместную работу. Одна из разновидностей такого цемента, которую часто недооценивают – *PCI Utilities*. Эффективно применяя *libpci*, эти утилиты дают возможность равно пользователям и системным скриптам прозондировать (а особо отважным – даже изменить) настройки оборудования и установки шины PCI. Возможно, вы ближе знакомы с просмотром устройств в псевдофайловой системе **/proc**, но тогда вы упускаете шанс насладиться некоторыми угощениями от *lspci*.

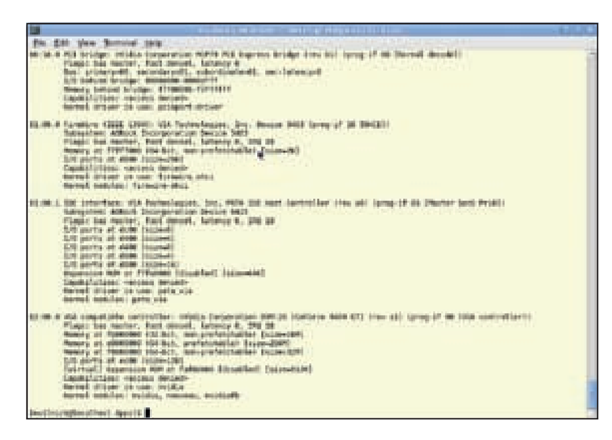
Запуск утилиты в оболочке напечатает список всех устройств, которые на данный момент обнаруживаются на шине PCI.

Смотрим вдумчиво

Вместо выдачи голых цифр эта удобная утилита проверит базу данных (в том числе в режиме онлайн), идентифицирует устройства и даст им более внятные названия. Пожелав узнать, на какой материнской плате вы работаете, или, по крайней мере, кто ее изготовил, вы сможете сделать это, не вскрывая корпуса.

А что вообще чудесно – введя переключатель **-k**, вы получите список драйверов ядра, загруженных для конкретного оборудования. Это ценное подспорье в обнаружении или диагностике неполадок видеокарты или какого-нибудь контроллере.

> «Уделите ей толику внимания — она спасет вас от проблем.»



› Режим подробного вывода *PCI Utilities*, среди прочего, даст вам список используемых драйверов ядра и ряда альтернатив.

PCI Utilities настолько полезны, что их включили почти все имеющиеся дистрибутивы, и, пожалуй, вы не ощущаете острой потребности в срочном обновлении в обход своего менеджера пакетов. Ну и ладно: последние релизы обновили базы данных, исправили ряд сбоев в отчетах о размере и провели ряд косметических улучшений. Но мы настоятельно рекомендуем уделить *PCI Utilities* толику внимания – изучите ключи командной строки; возможно, однажды это спасет вас от больших проблем.

Электронный САПР

Electric

Версия 8.11 **Сайт** www.staticfreesoft.com

Случалось ли вам прислушиваться к скрежету вашего CPU, ожидая, пока он обработает некие данные, и думать: «Спорим, у меня это вышло бы лучше?» Может, и нет, но с помощью этого изящного дизайнера полупроводников вы, по крайней мере, осознаете всю сложность сверхбольших интегральных схем. Если вы не планируете строить Atom или ARM нового поколения, можно разрабатывать и тестировать вещи попроще, или применить базовые инструменты и имитатор для глубокого изучения полупроводников N- и P-типа.

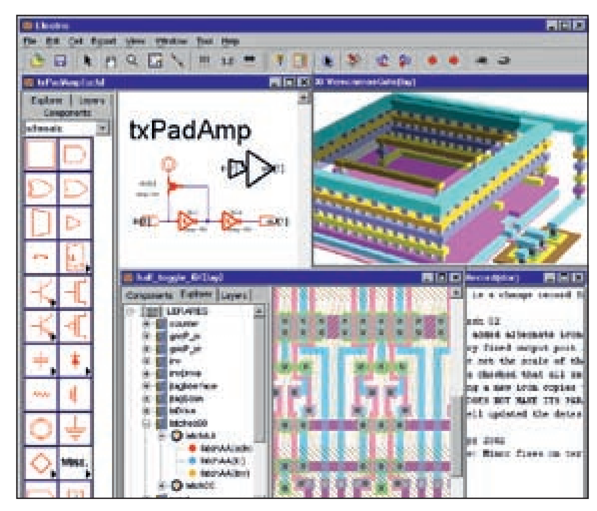
В терминологии *Electric*, каждая схема именуется «cell» – «ячейка», и можно разрабатывать, имитировать и переводить каждую отдельную ячейку в разные форматы и соединять их, создавая более сложные схемы. Среди имитаторов – Verilog и более знакомый (по крайней мере, пользователям открытого ПО) формат SPICE, хотя *Electric* предоставляет и собственные инструменты имитации, а также может генерировать файлы, пригодные для использования на коммерческих имитаторах.

Это кросс-платформенное приложение основано на Java, и, заполучив файл JAR, не трудитесь распаковывать его – просто запустите командой:

```
java -jar electric-8.11.jar
```

Буйство форматов

Из-за этого вам также придется смириться с несколько не-Linux'овым интерфейсом, но он все же неплох, как у всех приложений Java. Правда, не включено файлов для примера, но совместимых немало найдется в Сети. *Electric* поддерживает массу разнообразных форматов, в том числе SPICE, файлы Verilog, DXF CAD и множество родных форматов коммерческих инструментов VLSI, хотя к некоторым из них подходит весьма избирательно.



› В *Electric* есть инструменты почти для всех стадий разработки схем и компонентов, и даже показ в режиме 3D.

> «Другие Linux-средства для разработки схем не столь масштабны.»

В Linux есть и другие средства для разработки схем, но они далеко не столь масштабны. Иногда использование *Electric* достаточно сложно: в нем явно прослеживается наследие старых CAD-систем командной строки, но если вы жаждете разработать собственный чип или просто набросать дизайн печатной платы, вам его хватит.

Файловый менеджер

Worker

LINUX FORMAT Hottest Pick

Версия 2.17.6 **Сайт** www.boomerangsworld.de/worker

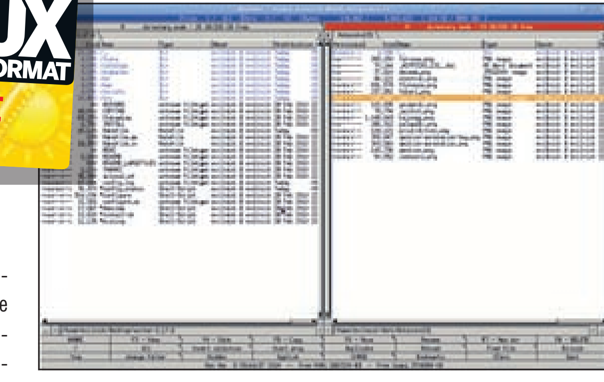
Поработайте в этом файловом менеджере чуть дольше пары минут, и вы поймете, как идет ему его имя [worker – трудяга, – *прим. пер*.]. Будь *Worker* грабителем, это был бы солидный грабитель: он честно подстерегал бы вас в темной аллее, чтобы приставить вам к горлу нож и потребовать кошелек, а не просто вырывал у вас из рук мобильник и давал деру, как один из этих нынешних молокососов.

Но *Worker* не опускается до банального грабежа, проникшись пролетарской славой своего имени. И в самом деле, его интерфейс, кажется, отлит из казенной стали на том же заводе, где ковалось и честное орало, и грозный меч – ну, в смысле, танк.

Вид двух панелей может вызвать ассоциации с *Midnight Commander*, но те, кто помнит 90-е на другой платформе, увидят, что Worker – не просто кивок в сторону Directory Opus: вы убедитесь в этом, взглянув на нижнюю панель кнопок. Нажмите на кнопку 'C' вверху слева, чтобы настроить работу, и вы увидите, что Worker ведет собственное маленькое досье типов файлов и связанных с ними действий. Настройте здесь свои собственные действия (вообще-то вам и придется это сделать, потому что выбор прочих действий, предоставляемых Worker, несколько темен) или добавьте новые типы файлов, согласно характеру вашей деятельности.

Будьте функциональны

И хотя *Nautilus*, *Dolphin* и *Konqueror* имеют определенные ниши в современном мире, трудно превзойти файловый менеджер, обладающий столь испытанным и проверенным дизайном, а также массовым воздействием на файлы и директории, скоростью и низким потреблением ресурсов.



› Добро пожаловать назад в 90-е. Да стоило ли отсюда уходить?

> «Пахарю файловой нивы Worker предложит именно то, что нужно.»

Worker хорош, но не совершенен – он не создает потоков, и это проблема, если нужно скопировать крупные файлы: придется дожидаться завершения операции и только потом приступать к другим задачам. Те, кто привык к претенциозным интерфейсам и ярким эффектам, тоже скоро заскучают, но вот простому пахарю, бороздящему файловую ниву, *Worker* предложит именно то, что нужно.

HotGames Развлекательные приложения

Пошаговая стратегия

Turious

Версия 0.2 **Сайт** http://bit.ly/cCVZAN



› Не особо волнующее приключение, но потенциал у игры все же есть.

Хотя хороших головоломок, не говоря уж об отстрелах инопланетян, в Linux хватает, количество стратегий здесь невелико. *Turious* – любопытный шаг в направлении создания игры, которая, возможно, в один прекрасный день созреет для употребления.

Механика крайне проста: обучайте на свои деньги фермеров или солдат. Фермеры создают благосостояние, и чем больше их обучаешь, тем большую прибыль они принесут в последующих ходах. Солдаты отражают нападения или пытаются одолеть соперника. Соблюдайте шаткое равновесие, вкладываясь в фермеров, чтобы добыть достаточно денег на создание боеспособной армии. Похоже, компьютерные игроки усвоили эту идею неплохо, но они сражаются также и друг с другом, и если как следует приглядеться к новостям в конце хода, можно понять, кто из них слабее, а кто сильнее.

Графические иллюзии

Эта новая версия *Turious* щеголяет графическим интерфейсом, который на деле не очень-то и нужен; но зато он создает впечатление настоящей игры. Ну, типа того. Щелкайте по строениям, чтобы выбрать команды из всплывающего меню. Подстройка чисел может показаться довольно нудным делом, потому что их нельзя просто ввести, но, по крайней мере, запоминаются значения, указанные в прошлый раз. Дисплей не анимирован, и он не изменяется в зависимости от происходящего, даже если на вас напали – чтобы узнать об этом, следите за всплывающим текстом в верхней части экрана. Единственная визуальная подсказка – значки, обозначающие других игроков, тускнеют, если те погибли.

> «Новая версия Turious щеголяет графическим интерфейсом.»

Turious поставляется в виде самораспаковывающегося архива, который можно запустить от имени root для установки. Для нормального хода игры также потребуются некоторые библиотеки SDL.

Мозаика

Kajongg

Версия 0.4.0 **Сайт** http://bit.ly/bFJuDr



› Дело не в очистке доски от парных костяшек. Научиться легко, но стать мастером –тут и жизни не хватит.

Некоторые люди могут твердить вам о своей любви к маджонгу, рассказывая, как находить сочетающиеся костяшки и как они попарно исчезают с доски. К сожалению, эти люди, вероятно, не знают, что такое маджонг на самом деле.

По некоторой причине это имя позаимствовали для бесконечной череды игр в стиле пасьянса, но настоящий маджонг больше похож на ремик. Надо собирать набор последовательностей или идентичных костяшек одной из трех мастей – палочек, колес или символов, или из особых мастей – ветров и драконов.

Игра идет по принципу «одну взял, одну отложил», пока один из четырех игроков не наберет нужную комбинацию. Есть множество вариаций и расширений правил, в том числе некие комбинации из особых костяшек, приносящие сразу много очков. На овладение основами игры не уйдет много времени, но стать в ней мастером – это совсем другой расклад.

Kajongg появился на свет под именем *Kmj*, но изменение названия совпало с важным релизом игры.

Веселье в компании

Помимо распределения костяшек между четырьмя игроками, эта новая версия 0.4 имеет также возможность игры с компьютером или по сети, серьезно увеличивающую вероятность нахождения соперников в любое удобное для вас время.

> «Изменение названия совпало с важным релизом игры.»

Помимо солидных функций редактирования правил, вы можете при желании убрать самые непонятные комбинации, или, если вы настолько продвинуты, добавить собственные. Здесь есть три различных масти костяшек, из которых можно выбирать, но нам показалось, что их не мешало бы слегка увеличить в размере – поскольку доска в маджонге квадратная, они умещаются на экране с трудом.

Помимо посредственной игры с компьютером, можно употребить *Kajongg* и для своих собственных развлечений с друзьями. Гремите, костяшки!

Системный монитор

XSensors

Версия 0.70 **Сайт** www.linuxhardware.org



› *XSensors* подробно информирует о работе системы, помогая избежать потенциальных катастроф.

Иногда небесполезно знать состояние своего оборудования – в частности, его температуру, как мы выяснили пару недель назад. Запуск тестовой машины со старым двухъядерным процессором Pentium (поколение до Core2: оно работает очень быстро и греется очень сильно) привел к тому, что все раскалилось, и мы умудрились спалить жесткий диск и часть материнской платы. А вот процессору хоть бы что.

Lm-sensors (который недавно подвергся обновлению) – это низкоуровневый драйвер и сервис, соединяющийся с компонентами материнской платы, способными отслеживать температуру и напряжение. И, как и у всего прочего в мире компьютеров, количество типов этих устройств стремится к бесконечности, поэтому охвачены не все из них; но, по крайней мере, большинство.

Lm-sensors собирает данные через соответствующие подсистемные шины (обычно интерфейс I2C), но перекладывает их отображение на внешний интерфейс. Запуск демона *Lm-sensors* позволяет вам считывать данные с системного монитора Gnome, или приложения *KSensors*, если вы работаете в KDE. Однако *XSensors* по ряду причин тоже достоин внимания.

Низкие издержки

Установка его простая и быстрая, а главное, не требует всех ресурсов рабочего стола KDE или Gnome. Это весьма полезно для тестовых машин и серверов, где было бы неразумно городить целый рабочий стол ради нескольких цифр. После его запуска появится список вкладок с распознанным оборудованием, подвергаемым мониторингу. Возможно, вам придется пощелкать по некоторым из них, чтобы найти нужную информацию. Данные всех мониторов выводятся в три колонки: напряжение, температура и скорость вентилятора. Отображение производится в виде псевдо-цифрового индикатора, и читать его достаточно легко.

> «Он не требует всех ресурсов рабочего стола KDE или Gnome.»

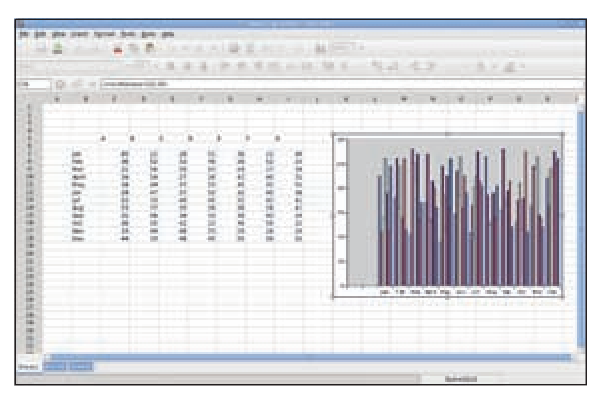
Озаботьтесь этим, не то вы рискуете утратить весь свой скомпилированный код HotPicks и каталог с дисками для Amiga.

Редактор электронных таблиц

Gnumeric



Версия 1.10.0 **Сайт** http://bit.ly/IYQw



› Возможно, *Gnumeric* и лишен взаимосвязанности приложений *OpenOffice.org*, зато быстрее и не так алчен к памяти.

В недрах пакета *Gnome Office* таится сверкающая жемчужина, и имя ей – *Gnumeric*. *Gnome Office* – это скорее свободная конфедерация приложений, возможно, не отличающаяся внутренней гармонией *KOffice* или объемного *OpenOffice.org*; но *Gnumeric* как отдельно взятый редактор электронных таблиц не имеет себе равных.

Он заслужил эту репутацию благодаря своей полноте и совместимости в качестве программы для работы с электронными таблицами. Он поддерживает нужные форматы файлов, имеет требуемые функции, быстр и прост в использовании и стабилен, как скала, даже при импорте «инородных» документов. Совместимость с *Excel* остается досадной необходимостью, и *Gnumeric* справляется с ней лучше других. Он также остается на высоте, когда доходит до поддержки ODF, хотя заморочки этого формата по-прежнему способны отложить полную переносимость на неопределенный срок.

Новая версия более совместима с *Excel* благодаря улучшению поддержки файлов XLL – кодового расширения для *Excel*.

Значение размера

Кроме того, размер таблиц больше не ограничен конечным числом строк и столбцов: все упирается в ресурсы вашего компьютера. Имеется также полезный набор новых формул и статистических функций.

Никто не оспаривает достоинств *OpenOffice.org* – он, безусловно, лучший, когда нужно быстро вставить график в презентацию или запустить электронную таблицу вместе с базой данных. Но, с другой стороны, если нужна просто электронная таблица, то *Gnumeric* быстрее и не ест всей доступной вам памяти. И среди прочих программ для работы с таблицами он явно лидирует по переносимости.

> «Gnumeric быстрее и не ест всей доступной вам памяти.»

Для тех, кто собрался скомпилировать *Gnumeric* прямо сейчас, в нем есть один недостаток: он требует самой последней версии библиотек *goffice*, а значит, придется сначала найти и установить их. Что касается плюсов, он отлично себя ведет и легко собирается, и вы получите самые новые функции задолго до любого из дистрибутивов – вряд ли эта серия 1.10.x станет частью промежуточного обновления для большинства из них.

Безопасность

Jailkit

Версия 2.11 **Сайт** http://olivier.sessink.nl/jailkit

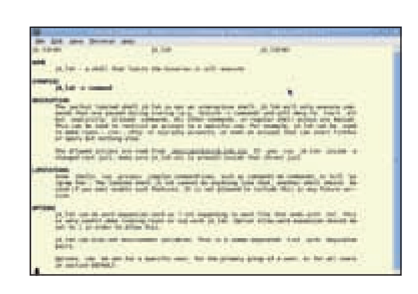
Никакое время, потраченное на безопасность вашей Linux-машины и ваших систем, не является чрезмерным. Стандартная практика заключается в построении карантина *chroot* для приложений, которыми можно манипулировать, чтобы предоставить удаленному пользователю права root.

Команда **chroot** позволяет администратору системы настроить среду, предоставив только инструменты, необходимые для процесса. Обычно такой процедуре подвергаются Web- и FTP-серверы и прочие программы, которые используют системные учетные записи и предоставляют доступ к файловой системе.

Настроить директорию *chroot* в общем несложно, но система скриптов *Jailkit* ускоряет и упрощает этот процесс; а главное, не забывает настроить права доступа и сделать все необходимое для обеспечения безопасности и надежности среды *chroot*.

Здесь даже есть скрипт проверки созданной вами среды и выдачи отчета о проблемах. Однако учтите: плохо настроенный карантин хуже, чем его полное отсутствие. Если пользователь получит доступ на запись в корне директории карантина или важных папок в ней, например, **/etc/**, он может дорваться до прав администратора.

Jailkit поставляется с набором скриптов, и в данной версии каждый из них снабжен исчерпывающей **man**-страницей. Правда, для успешного пользования программой вам придется все их читать и вносить изменения в файлы настройки. Но если требуется создать несколько карантинов *chroot*, затраченные усилия окупятся.



› Данная версия *Jailkit* идет с оболочкой с ограниченным числом команд.

Виртуальная клавиатура

Florence

Версия 0.4.6 **Сайт** http://florence.sourceforge.net

Даже в наш век сенсорных экранов и акселерометров самонужнейшим устройством при работе с компьютером остается клавиатура. А если ее нет? Или есть, но ею нельзя воспользоваться?

Экранные клавиатуры возникли едва ли не одновременно с мышью – было бы некое устройство ввода, а нажатие на клавиши можно воспроизвести практически в любой программе.

Florence продвинута чуть более некоторых ее соперников: ее можно настраивать и менять масштаб. Для работы с приложением, где задействованы функциональные клавиши или кнопочная панель, ускоряющая ввод чисел, во *Florence* есть дополнительные клавиши. При изменении размера окна изменятся и размеры клавиш; это очень удобно, если у вас огромный монитор, слабое зрение или вы любитель поиграть с клавиатурой во весь экран.

Опций настройки уйма. Можно велеть *Florence* выходить на первый план при всяком нажатии на нечто, требующее ввода текста; выбрать один из трех стилей клавиш или масштаб для выделенной клавиши; настроить прозрачность; убрать оконную рамку. Подобрать все это правильно, чтобы *Florence* стала подмогой, а не помехой – задача, быть может, не из самых легких, но она стоит потраченного времени.

Версия 0.4.6 должна входить в репозитории вашего дистрибутива, а исходники имеются на DVD. Системные требования невысоки, но понадобятся обычные библиотеки разработчика Gnome. LXF

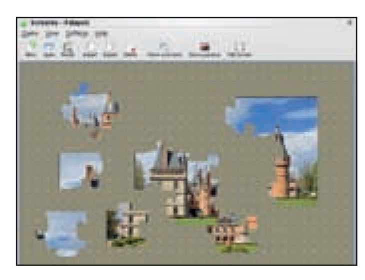


› Клавиатура *Florence* настраивается и масштабируется под ваши нужды.

Также вышли

Новые и обновленные программы, тоже достойные внимания…

Palapeli 1.0
Приложение-мозаика, сохраняющее все на ваш диск в реальном времени.
http://bit.ly/cKUUoK



› **В *Palapeli* кусочки мозаики не привязаны к воображаемой сетке.**

Milax 0.5
OpenSolaris раньше работал на больших железяках, а теперь работает в режиме Live с визитки.
www.milax.org

VirtualBox 3.1.6
Отличная программа виртуализации от Sun, исправившая массу ошибок.
www.virtualbox.org

Ecasound 2.7.1
Мультитрековый аудиопроцессор исправил ряд недочетов.
http://eca.cx/ecasound

Bleachbit 0.7.3
Пресеките бардак, который устраивают ежедневно используемые приложения.
http://bleachbit.sourceforge.net

Wine 1.1.41
Улучшенная поддержка реестра и программ-установщиков для не-эмулятора Windows.
www.winehq.org

Zim 0.44
Создайте wiki на своем рабочем столе и храните заметки, списки нужных дел и многое другое.
http://zim-wiki.org

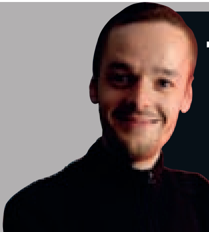
LCDtest 1.18
Инструмент на основе SDL, чтобы отмечать на вашем ЖК-дисплее битые пиксели.
http://bit.ly/bkiQi3

DeVeDe 3.16.4
Графический транскодер исправил многопроходовое кодирование и аудио-битрейт.
www.rastersoft.com/programas/devede.html

Giggle 0.4.96
Разработка этого Git-менеджера с GUI очнулась.
http://live.gnome.org/giggle

На диске

Все, что вам нужно, чтобы скоротать время до выхода LXF132...



Читайте, чтобы найти всю информацию, которая вам понадобится для использования этого DVD! Если вы новичок в Linux, откройте index.html на диске (Сторона 1) и перейдите в раздел Справка: там вы найдете мини-учебник, дающий ответы на вопросы справа.

Майк Сондерс, редактор диска
mike.saunders@futurenet.com

- » Что такое Linux?
- » Что такое дистрибутив?
- » Загрузка ПК с DVD
- » Разбиение жесткого диска на разделы
- » Навигация по файловой системе
- » Использование командной строки
- » Установка программ

Офисный пакет

OpenOffice.org 3.2

Некоторое время назад самый мощный офисный пакет для Linux обновился до версии 3.2, и вы можете прочитать об основных изменениях в нашем обзоре (на стр. 11). *OpenOffice.org* всегда имел репутацию увесистого монстра с затянутым временем загрузки, но с выходом серии 3.x оно значительно сократилось, и если вы используете более старый релиз, рекомендуем обновиться.

В разделе **Рабочий стол/OpenOffice.org** на LXFDVD вы найдете два сжатых архивных файла: один заканчивается на **-deb.tar.gz** а другой – на **-rpm.tar.gz**. Файл **-deb** следует использовать для Ubuntu, Debian и прочих дистрибутивов на базе Debian, а файл **-rpm** – для Fedora, Mandriva, OpenSUSE и других систем, применяющих формат RPM. Перед началом установки удалите все имеющиеся версии *OpenOffice.org* с помощью менеджера пакетов вашего дистрибутива, затем скопируйте соответствующий файл с DVD в свою домашнюю директорию, откройте терминал и следуйте приведенным ниже инструкциям.

Для систем на базе Ubuntu/Debian:

1. Введите **tar xfv OOo_3.2.0_LinuxIntel_install_ru_deb.tar.gz**, чтобы распаковать архив.
2. Введите **cd OOO320_m12_native_packed-1_ru.9483/DEBS/** для перехода в созданный каталог.
3. Переключитесь на пользователя root (администратора) командой **sudo bash** или **su** (в зависимости от вашего дистрибутива).
4. Введите **dpkg -i *.deb**, чтобы установить пакеты в вашу систему.

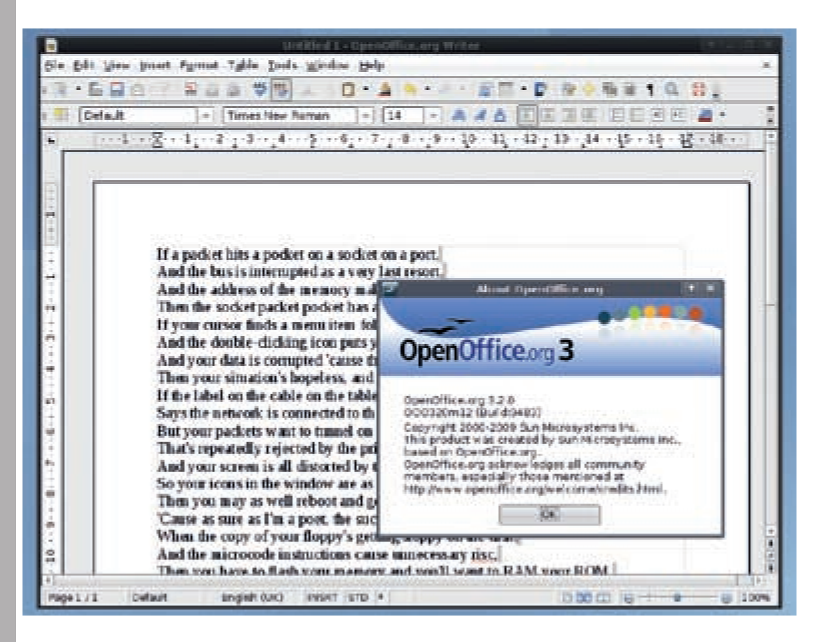
Для систем на базе RPM:

1. Введите **tar xfv OOo_3.2.0_LinuxIntel_install_ru.tar.gz**, чтобы распаковать архив.
2. Введите **cd OOO320_m12_native_packed-1_ru.9483/DEBS/** для перехода в созданный каталог
3. Переключитесь на пользователя root (администратора) командой **sudo bash** или **su** (в зависимости от вашего дистрибутива).
4. Введите **rpm -i *.rpm**, чтобы установить пакеты в вашу систему.

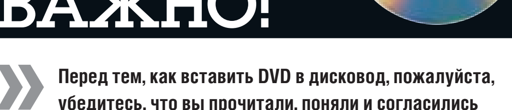
Установив пакеты, можете запустить *OpenOffice.org*, командой

```
/opt/openoffice.org3/program/soffice
```

В зависимости от вашего рабочего стола или менеджера окон, вы, возможно, обнаружите новые пункты для разных компонентов пакета (*Writer*, *Calc* и т.д.) в меню



› **Скорость загрузки *OpenOffice.org 3.2* выше, чем у предыдущих версий.**



» Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим.

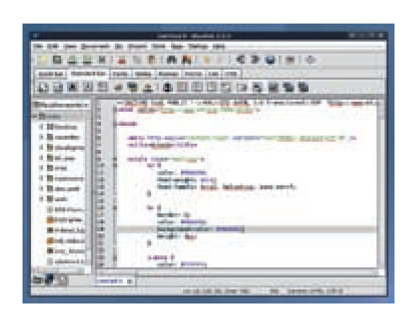
Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция *Linux Format* не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru для получения содействия.



› *Bluefish* теперь снабжен справкой, автодополнением кода и автовосстановлением.

приложений. Если нет, просто создайте ярлык для приведенной выше команды – документация вашего рабочего окружения подскажет, как это сделать. Ради экономии места можно также удалить файл **.tar.gz** и каталог, созданный при его распаковке.

Редактор HTML
Bluefish 2.0

Один из самых популярных HTML-редакторов с открытым кодом, *Bluefish*, претерпел значительные изменения со времен серии 1.x, и последний релиз 2.0 переполнен превосходными новыми функциями. Самые примечательные среди них:

» Автозаполнение тэгов HTML и атрибутов CSS.
» Удобная текстовая подсказка для атрибутов CSS. Она всплывает, когда вы набираете текст или задерживаете курсор мыши над атрибутом.
» Встроенная проверка орфографии (которая, понятное дело, игнорирует тэги).
» Фрагменты [Snippets] – функция, сменившая старую реализацию настраиваемого меню. Создавайте текст «до» и «после», и фрагмент при двойном щелчке по нему будет размещен на вашей web-странице, с курсором посередине.
» Таблица символов для просмотра HTML-сущностей (модуль расширения).
» Восстановление документов – в случае сбоя *Bluefish* просто перезапустите программу, и ваш код останется в целости и сохранности.

Есть и другие новые опции и инструменты, и в целом это весьма впечатляющий релиз – браво, разработчики! Если вы постоянно возитесь с сайтами, обходясь обычным простеньким текстовым редактором, дайте *Bluefish* шанс – его горячие клавиши и подсказки способны вызвать невероятный всплеск вашей производительности.

Вы найдете *Bluefish* на LXFDVD в разделе **Интернет**. Там имеются пакеты для Ubuntu и Fedora, которые можно установить через менеджер пакетов вашего дистрибутива (двойной щелчок по ним в вашем файловом менеджере тоже должен сработать). Если не получится, или если у вас другой дистрибутив, у нас есть также исходный код. Скопируйте **bluefish-2.0.0.tar.gz** в свою домашнюю директорию, откройте окно терминала и введите

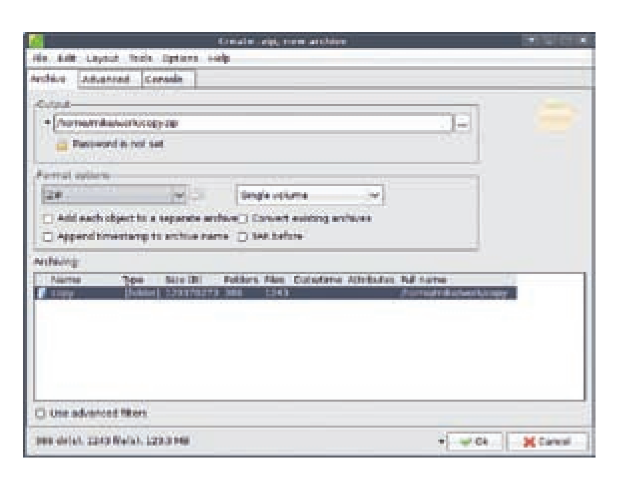
```
tar xvf bluefish-2.0.0.tar.gz
cd bluefish-2.0.0/
./configure
make
```

Исходный код распакуется; произойдет переход в полученную директорию и компиляция. Если у вас будут отсутствовать некоторые зависимости, вам об этом сообщат; найдите их с помощью менеджера пакетов вашего дистрибутива. Затем переключитесь на пользователя root (администратора) с помощью **su** или **sudo bash**, затем введите **make install**. Наберите **exit**, чтобы вернуться в учетную запись обычного пользователя, а затем **bluefish**, чтобы запустить приложение. (Если у вас возникнут проблемы, загляните в раздел **Справка/Новичку_в_Linux** на LXFDVD: там имеется развернутое руководство по сборке из исходных текстов и использованию командной строки.)

Архиватор
PeaZip 3.0

Возможно, вы уже слышали о *PeaZip*; а если нет, то это – потрясающе мощная программа-архиватор. Вы можете сжимать файлы в 12 форматах (**.zip**, **.gz**, **.bz2**, **.7z**, **.sfx**, **.paq** и др.), и распаковывать аж 118 форматов, в том числе те, о которых вы, возможно, и слыхом не слыхивали. Если вы найдете в Интернете файл, который поставит в тупик ваши обычные инструменты, *PeaZip* почти наверняка будет знать, что делать, и мы рекомендуем всем обзавестись им.

Версия 3.0 отличается более быстрой проверкой архивов, усовершенствованным сообщением об ошибках и улучшенным интерфейсом. На диске она имеется в форматах **.rpm** (Fedora, Mandriva, OpenSUSE) и **.deb** (Ubuntu, Debian и т.п.) или в виде универсального **.tgz**, который можно распаковать в корневую директорию (**/**). Стандартная версия базируется на инструментарии *GTK* и по этой причине отлично сочетается с рабочими столами Gnome и *Xfce*, но если у вас KDE, попробуйте сборку в директории *Qt*.



› *PeaZip* умеет распаковывать 118 разных видов архивов.

Всегда на диске
Советы и Ответы

Упомянем также пару постоянных разделов LXFDVD, которые не стоит обходить вниманием: во-первых, это Архив ответов в разделе **Справка**. На протяжении нескольких лет мы собирали для вас обширную коллекцию проблем Linux и их решений из раздела Ответы нашего журнала, и вы можете просмотреть их в браузере, с возможностью полнотекстового поиска. Если вдруг в вашей системе притаился гремлин, загляните сюда.

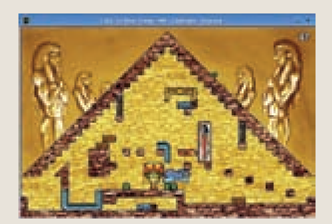
А еще здесь есть специальный документ в помощь новичку. Если вы делаете свои первые (пусть и не робкие) шаги в свободной ОС, и предложения вроде «установите пакет», «распакуйте и скомпилируйте исходные тексты», «перейдите в терминал» и подобные вызывают у вас недоумение – обратитесь к разделу **Справка/Новичку в Linux**.

И это еще не все!
Пять классных игр

На сей раз это в основном головоломки, и первая из них – *Chroma*, многоцветная абстрактная игра. Вы перемещаете два голубых комка по экрану, собирая звездочки. Но ваш путь преграждают треугольники разных цветов, и они тоже перемещаются по мере того, как вы очищаете поле от препятствий. Как эти треугольники взаимодействуют друг с другом, в самом начале игры не объясняется: ваша задача – сообразить, что они делают, и в соответствии с этим планировать свои действия по сбору звездочек.

Еще у нас имеется новый релиз *Fish Fillets NG*, превосходной и отлично сделанной головоломки, в которой вы управляете парой водяных жителей и должны выпутаться из 70 нелегких ситуаций. Рыбы болтают и вышучивают друг друга, обеспечивая вам немало веселых минут, которые отвлекут вас от мыслей о налогах. Игра очень теплая, интересная и благодарная – это одно из наших любимых развлечений в Linux.

А вот *MoonPy* – свободный римейк *Moonbase Commander*, пошаговой стратегии 2002 года. Вы стартуете с базы на планете и каждым своим ходом атакуете противников или добавляете оружия и строений на своей базе. Затем идет *GearHead 2*, *Rogue*-подобная игра: в ней фигурируют гигантские роботы. *Octaspire Crates* – увлекательная 3D-головоломка; к ней надо привыкнуть, но она снабжена руководством, которое поможет вам сделать первые шаги.



› *Fish Fillets NG* обаятельна и увлекательна — и дьявольски сложна на последних уровнях.

На диске

Новые дистрибутивы и обучающие материалы...

Дистрибутив Linux

MOPS Linux 7.0

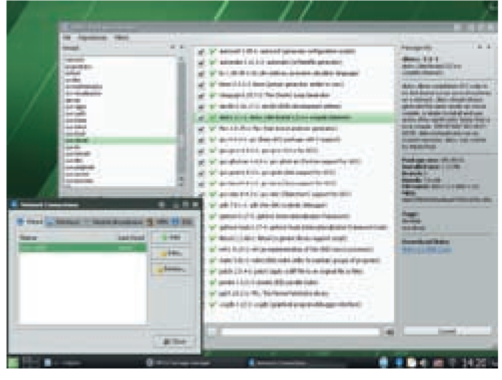
Если вы пользуетесь Linux долее месяца, то не могли не слышать о Slackware. Наверняка вы знаете, что это почтенный дистрибутив «старой школы» с текстовым установщиком и достаточно скромным по современным меркам официальным репозиторием, поддерживаемый одним человеком – Патриком Фолькердингом [Patrick Volkerding].

Все это верно, но помимо анахронизмов Slackware имеет и несомненные плюсы – он легок, у него простой и ясный дизайн, который легко перекроить под себя. Неудивительно, что сообщество (в том числе русскоязычное) предприняло множество попыток «осовременить» дистрибутив и расширить диапазон доступных для него приложений. Одной из самых успешных, на наш взгляд, стал MOPSLinux.

Изначально разработанная московским НПО «Сеть» для внутренних нужд, система быстро завоевала себе множество преданных поклонников. Версия 7.0, выпущенная в начале апреля – это действительно шаг вперед. Загрузчиком инсталляционного DVD теперь является *Grub2* (с интегрированными функциями *SuperGrubDisk*). Был значительно переработан установщик – он по-прежнему текстовый, но больше не выглядит гостем из прошлого. Все пакеты MOPS теперь поддерживают PAM (который, как известно, отсутствует в Slackware), что позволяет легко менять схемы аутентификации.

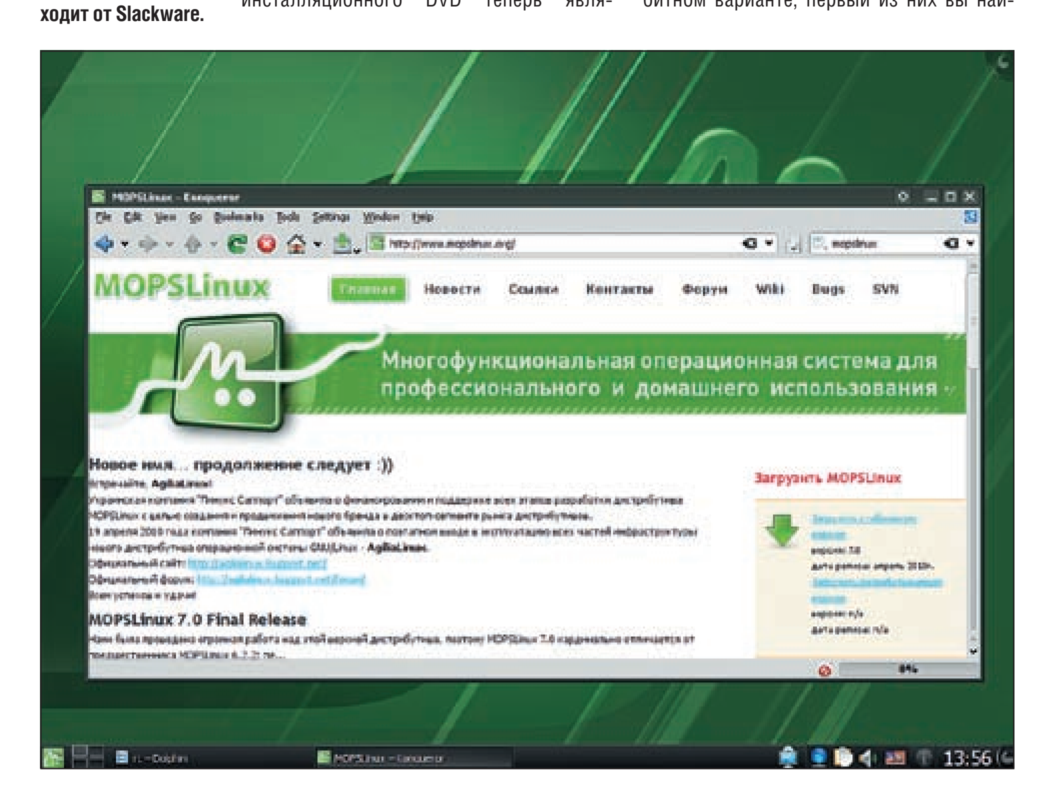
Изменения затронули и рабочий стол. Во-первых, их стало несколько (KDE 4.4, LXDE 0.5, *Xfce*); во-вторых, сетями теперь управляет *NetworkManager*, а за плавную загрузку системы отвечает *Plymouth*. Усовершенствованный менеджер пакетов умеет различать альтернативы – если вам требуются различные сборки одного и того же приложения (скажем, с поддержкой KDE и без оной), то это больше не проблема.

MOPSLinux 7.0 выпускается в 32- и 64-битном варианте; первый из них вы найдете на второй стороне **LXF**DVD. Просто загрузитесь с нее и следуйте экранным подсказкам.



› **В MOPS есть графические инструменты для управления пакетами и сетью.**

› **С виду MOPS Linux 7.0 ничем не отличается от других современных дистрибутивов: даже не верится, что он происходит от Slackware.**



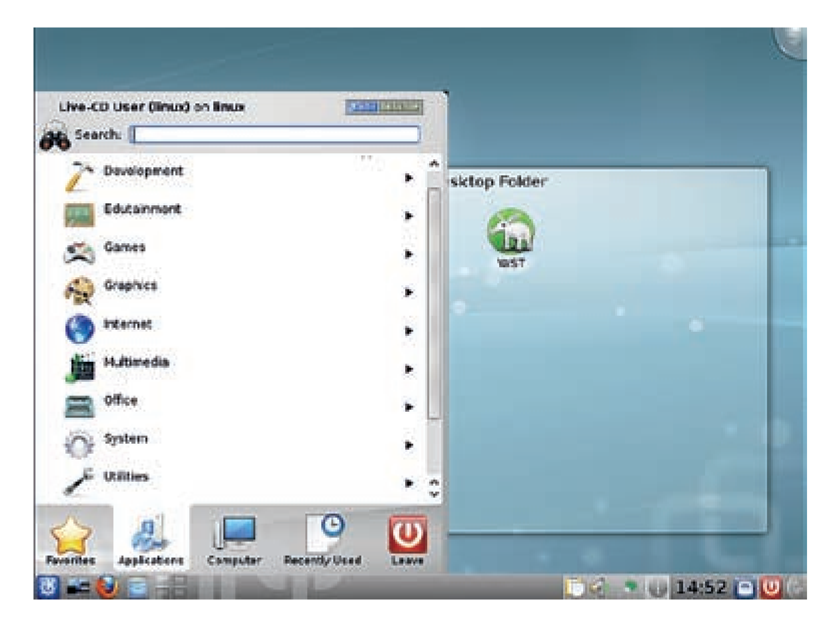
К сожалению, все хорошее тоже имеет конец. НПО «Сеть» сообщила, что 7.0 будет последним релизом MOPSLinux, выпущенным компанией. Однако к моменту, когда пишутся эти строки, упавшее знамя подхватила украинская фирма «Линукс Саппорт» (http://lsupport.net) – такое едва ли случилось бы, будь MOPS проприетарным продуктом. Дистрибутив должен появиться под новым брендом уже к тому моменту, когда вы увидите этот номер журнала – и мы надеемся, что продолжение будет достойным оригинала.

Live-дистрибутив Linux

KDE 4.4 Live

После главного релиза 4.0 KDE 4 прошел долгий путь. Многие пользователи – в том числе и давние поклонники этой среды рабочего стола – сочли 4.0 довольно незрелым: в нем отсутствовал ряд функций, зато присутствовал ряд проблем, которые следовало устранить. Защищаясь, разработчики KDE утверждали, что 4.0 создан для программистов и продвинутых пользователей, а остальные пусть подождут версии 4.1, которая исправит некоторые недостатки. Ну, а 4.4 – очень мощная и полнофункциональная среда. Более подробная информация – в нашем обзоре на стр. 10.

В прошлом месяце мы включали исходный код KDE 4.4 для тех, кто не боится собирать крупные программные проекты из исходных текстов. В этом выпуске у нас



› KDE 4.4 — лучший релиз 4.x, и он рвется к бою прямо с нашего DVD.

специальный Live-дистрибутив, который сразу загружает рабочий стол KDE 4.4 – вы можете работать с ним прямо с диска, и не надо ничего устанавливать. Если вы – постоянный пользователь KDE, то просто обязаны его попробовать, а если вы пользуетесь Gnome/*Xfce*/*FVWM*, познакомьтесь с ним, чтобы быть в курсе новинок этой графической среды.

Для запуска KDE Live, загрузите свой компьютер с первой стороны **LXF**DVD. (То есть запустите компьютер, установив диск в привод – если перед вами не появится загрузочное меню, придется, возможно, изменить настройки BIOS, чтобы компьютер загружался с DVD, а не с жесткого диска.) В загрузочном меню выберите KDE 4.4 Live. Если при загрузке возникнут проблемы, перезапустите машину и выберите в меню пункт Safe Mode.

Через пару мгновений появится рабочий стол, и вы сможете приступить к знакомству с новыми функциями и имеющимися программами. Меню приложений откроет кнопка K внизу слева, а красная кнопка внизу справа выключает компьютер. Пользователи OpenSUSE заметят, что данный дистрибутив основан на SUSE, так что здесь присутствует *Yast* вместе с обычным инструментарием.

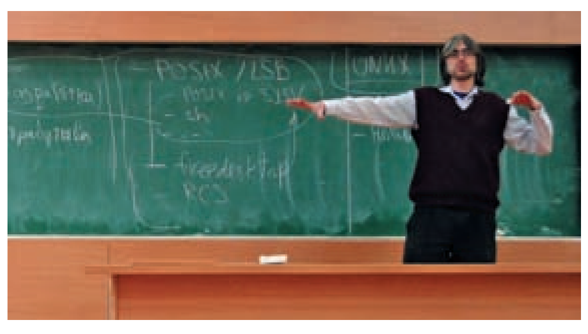
Обучение

Видеолекции по ОС и сетям

Раз вы выписываете (или покупаете) *Linux Format*, значит, вам нравится узнавать что-то новое о свободном ПО и мире вокруг него, правильно? Печатные и онлайн-ресурсы хороши всем, кроме одного: они по определению статичны. А как хорошо было бы не только увидеть, но и услышать – живого специалиста, пусть даже в записи.

По вашим просьбам на DVD этого месяца размещены аудио- и видеолекции по компьютерным сетям и операционным системам, читавшиеся Георгием Курячим в рамках проекта UNeex (http://uneex.ru/LecturesCMC) на факультете вычислительной математики и кибернетики Московского государственного университета (ВМИК МГУ) в 2009 году. Данные материалы были на компакт-дисках, врученных всем участникам V конференции «СПО в Высшей Школе» (http://freeschool.altlinux.ru/?p=2658), и поскольку они выпущены по лицензии CC-BY-SA 3.0, мы тоже получили возможность опубликовать их на **LXF**DVD. К сожалению, нам пришлось изменить компоновку: часть лекций вы найдете на стороне 1, а часть – на стороне 2 (в обоих случаях в разделе **Журнал**). Лекции охватывают сопровождение пакетов в Linux (это, пожалуй, один из наиболее общественно-полезных способов для желающих принять участие в развитии СПО) и межсетевые экраны. Видеоматериалы представлены в Ogg Theora, аудио – в Ogg Vorbis, и они должны воспроизводиться в любом дистрибутиве без установки проприетарных кодеков.

В заключение процитируем одного из участников сети Habrahabr, скрывающегося под ником gumanoed: «Георгий великолепно рассказывает о довольно сложных понятиях. Обычно лекции разбавляются шутками и авторскими афоризмами, что делает их еще более интересными и запоминающимися. … Это именно то, что я ждал услышать в институте, но так и не дождался за все время обучения». LXF

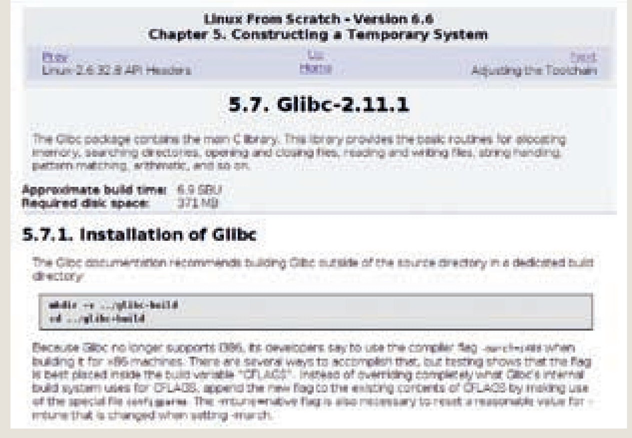


› Лучше один раз увидеть, чем сто раз прочитать: теперь вы тоже можете посетить курсы по Linux.

Отточим навыки Linux From Scratch 6.6

А вот кое-что, во что с радостью вонзят зубы пользователи Linux среднего и продвинутого уровня: проект *Linux From Scratch*. Если вы уже немного поработали в Linux, вы, вероятно, разобрались в этой системе и можете решить большинство возникающих проблем. Благодаря теме этого номера вы получите дополнительные знания о том, как все части операционной системы функционируют как единое целое.

Но если вы хотите сделать следующий шаг и собрать Linux с нуля, то вам прямая дорога в *Linux From Scratch* (*LFS*). Это сказочно подробная и отлично написанная электронная книга, описывающая все шаги, от создания раздела под новую систему до компиляции основных пакетов и написания загрузочных скриптов. Читая ее, вы узнаете, какие программы зависят от других, где что располагается в файловой системе и как исправлять или обновлять основные компоненты в будущем. Короче, если вы пройдете весь процесс *Linux From Scratch*, вы с полным правом сможете именоваться Linux-Гуру.



› Станьте истинным Linux-гуру с LFS.

На **LXF**DVD, в каталоге **Distros/Linux_From_Scratch** лежит новый релиз 6.6 книги *LFS* в форматах PDF и HTML, а также tar-архив с исходными пакетами для компиляции основной системы – *Bash*, *glibc*, *GCC* и т.д. Их хватит, чтобы пройти весь путь от голой оболочки до полноценной ОС Linux.

Поверх системы LFS можно еще многое чего построить. На www.linuxfromscratch.org есть и другие проекты, типа BLFS (Beyond Linux From Scratch), нацеленного на настройку *X Window System* и графических приложений, или HLFS, с акцентом на максимум безопасности.

Информация о диске

Внимательно прочтите это перед тем, как использовать LXFDVD!

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- **имя_программы-1.0.1.i386.rpm** – вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- **имя_программы-1.0.1.i386.deb** – такой же пакет, но уже для Debian;
- **имя_программы-1.0.1.tar.gz** – обычно это исходный код;
- **имя_программы-1.0.1.tgz** – тот же файл, что и выше по списку: «tgz» – это сокращение от «tar.gz»;
- **имя_программы-1.0.1.tar.bz2** – тот же файл, но сжатый bzip2 вместо обычного gzip;
- **имя_программы-1.0.1.src.rpm** – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- **имя_программы-1.0.1.i386.FC4.RPM** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- **имя_программы-1.0.1.ppc.Suse9.rpm** – двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- **имя_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru



DVD-приложение к журналу Linux Format №5 (131). Май 2010

Содержание

LINUX FORMAT

Сторона 1

РАБОЧИЙ СТОЛ

Celestia – расширения для настольного планетария
Digikam – продвинутое приложение для управления фотоколлекцией
F-Spot – менеджер фотоколлекции
FPM2 – программа для хранения паролей
Fotoxx – свободная программа для несложного редактирования изображений
KPhotoAlbum – утилита для просмотра, комментирования и поиска изображений
Nano – текстовый редактор
OpenOffice.org – офисный пакет
PeaZip – архиватор
Rawstudio – RAW-конвертер
RawTherapee – программа для обработки цифровых фотографий
Rosegarden – профессиональный ауио- и MIDI-секвенсор
TaskJuggler – инструмент для управления проектами
Tellico – менеджер коллекций для KDE
VueScan – программа для сканирования документов и фотографий
X-tile – апплет для склеивания окон

РАЗРАБОТКА

Botan – библиотека криптографических алгоритмов на C++
Chicken – компилятор языка Scheme
Gambas – графическая среда разработки для Basic-подобного языка
Geany – легкая среда разработки
PHP – интерпретатор языка PHP
Vala – язык программирования для Gnome

ДИСТРИБУТИВЫ

KDE Live – живой дистрибутив с последней версией среды KDE
LFS – комплект для сборки дистрибутива
NetBSD – свободная, безопасная и переносимая Unix-совместимая операционная система

HOTPICKS

Electric – система автоматизированного проектирования
Florence – расширенная масштабируемая виртуальная клавиатура для Gnome
FUBAR – инструмент для бесконечного масштабирования изображений
Gnumeric – электронная таблица
Jailkit – набор утилит для ограничения пользовательских учетных записей
Kajongg – классический маджонг для четырех игроков
PCI Utilites – коллекция программ для опроса и управления настройками PCI-устройств
QMotion – программа для распознавания жестов
Turious – пошаговая стратегическая игра
Worker – двухпанельный файловый менеджер
XSensors – программа для отображения информации с сенсоров материнской платы

ИГРЫ

Chroma – абстрактная головоломка
Fish Fillets – логическая игра
GearHead 2 – ролевая игра
MoonPy – открытая переделка классической стратегической игры Moonbase Commander
OCTASPIRE CRATES! – стимулирующая и развлекающая трехмерная головоломка

ИНТЕРНЕТ

Bluefish – мощный HTML/XHTML/XML-редактор
Gnome Gmail – утилита для добавления поддержки Gmail в Gnome
KDE Personal WebServer – веб-сервер для пользовательских нужд
Kwooty – программа для скачивания бинарных данных из usenet
Pino – Twitter-клиент
Uget – менеджер закачек

СИСТЕМА

ClamAV – открытый антивирус
DOSBox – эмулятор DOS
FindRepe – утилита для поиска дубликатов
JNode – исследовательская ОС на Java
Vpnc – KDE-интерфейс для различных vpn-клиентов
mandagelogs – программа для обработки файлов журналов

Сторона 2

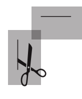
MOPS Linux 7.0 – последняя версия основанного на Slackware дистрибутива
Аудио- и видеоматериалы семинаров UNИX, проходящих в МГУ

Пожалуйста, перед использованием данного диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИИ? Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕФЕКТНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь пожалуйста по адресу disks@linuxformat.ru

Тираж изготовлен ООО «Уральский электронный завод». 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПТР ВАФ 77-15.



Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков: подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика.

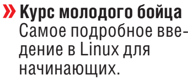


Пропустили номер?

» Мир свободного ПО богат и разнообразен, а потому далеко не все можно вместить в рамки одной статьи. Linux Format обходит эту проблему, публикуя серии статей по самым актуальным вопросам, но что делать, если вы поймали интересующий вас материал на середине? Обратитесь в Линуксцентр по адресу www.linuxcenter.ru и закажите желаемый номер журнала! Он доставляется как в печатной, так и в электронной форме, поэтому с момента открытия браузера и до получения нужного вам выпуска **LXF** может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:

LXF128
Февраль 2010

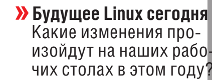


» **Курс молодого бойца** Самое подробное введение в Linux для начинающих.
» **Десятиминутки** Девять проектов, которые отнимут у вас не больше полутора часов.
» **Testdisk** Восстановите случайно уничтоженные разделы и верните потерянные данные.
» **Clutter** Создавайте приложения для платформы Moblin или для Gnome 3.0.

LXFDVD: Fedora 12 и подшивка **LXF** за первую половину 2009 года!

Печатная версия:
http://shop.linuxformat.ru/lxf_128/
Электронная версия в формате PDF:
http://shop.linuxformat.ru/elxf_128/

LXF129
Март 2010

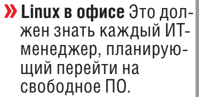


» **Будущее Linux сегодня** Какие изменения произойдут на наших рабочих столах в этом году?
» **Ядро ждет помощи от ВАС** Впишите свое имя в списки разработчиков Linux.
» **Android** Создание приложений для мобильной платформы Google.
» **gPhoto и PTP** Управляйте цифровой камерой с компьютера.

LXFDVD: Linux Mint 8, FreeBSD 8.0, Calculate Linux 10.2, а также книги в PDF

Печатная версия:
http://shop.linuxformat.ru/lxf_129/
Электронная версия в формате PDF:
http://shop.linuxformat.ru/elxf_129/

LXF130
Апрель 2010



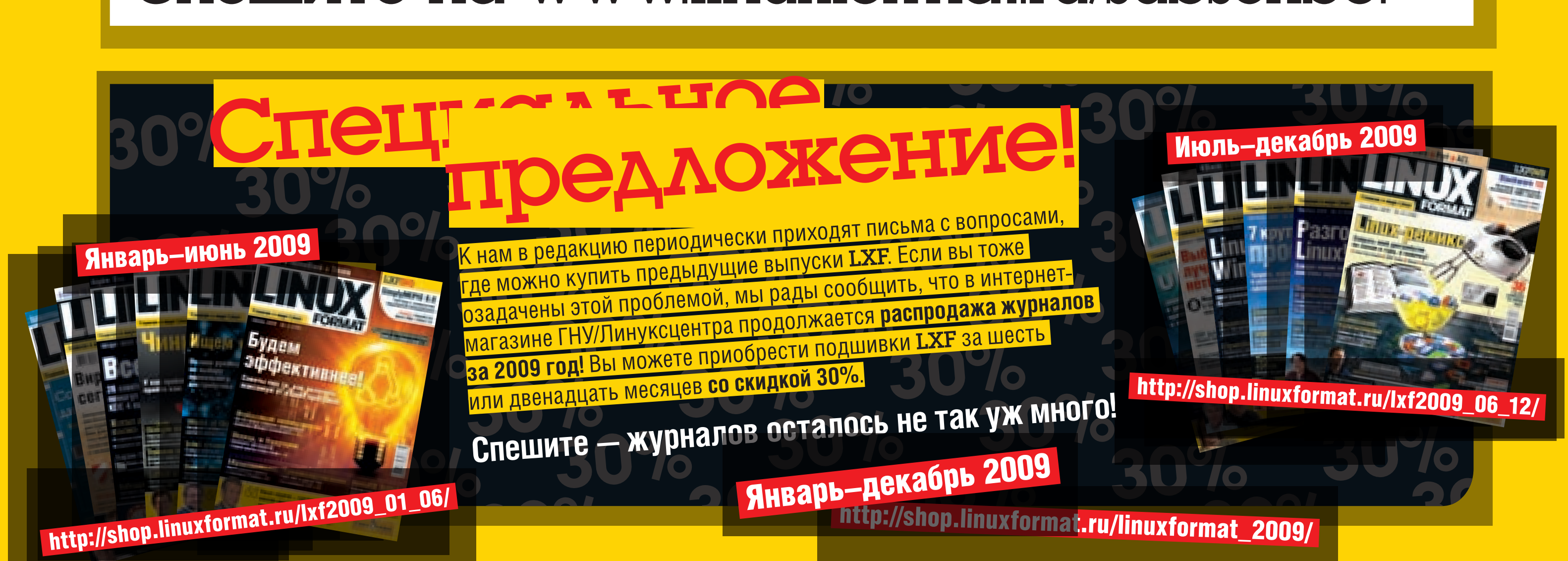
» **Linux в офисе** Это должен знать каждый ИТ-менеджер, планирующий перейти на свободное ПО.
» **Звук** Как работает аудиоподсистема в свободной ОС?
» **Google Go** Освойте новый язык для разработки многопоточных приложений.
» **OpenVZ** Быстрая виртуализация Linux в Linux.

LXFDVD: мини-дистрибутивы, Hymera Open, Pardus 2009.1

Печатная версия:
http://shop.linuxformat.ru/lxf_130/
Электронная версия в формате PDF:
http://shop.linuxformat.ru/elxf_130/

Ну, а если вы хотите быть уверенными, что не пропустите ни один номер журнала – оформите подписку! Помните, что все подписавшиеся на печатную версию журнала через www.linuxcenter.ru получают электронную версию в подарок!

Спешите на www.linuxformat.ru/subscribe!



Обмен опытом и передовые идеи по использованию свободного ПО в образовании

Поехали!



Хотите показать ученикам Марс с близкого расстояния? Отправить их в пояс астероидов? Нет ничего проще – **Александр Бикмеев** расскажет, что вам для этого потребуется.



Наш эксперт

Александр Бикмеев
Кандидат физико-математических наук, доцент и штатный преподаватель авиационно-технического вуза. А какой физик не мечтал быть астрономом?

В нашем журнале уже не раз появлялись обзоры свободных программ, применяемых в астрономии (например, в **LXF77** и **LXF105**). Множество статей на эту тему можно найти и в Сети. Однако хватит говорить о том, что существует в этом мире – пора начать исследовать этот мир! Другими словами, мы предлагаем вам взять в руки одно из множества приложений и стать гидом по Вселенной.

В мире проприетарного программного обеспечения (ПО) безусловным лидером среди образовательных и исследовательских астрономических программ среднего уровня (по крайней мере, в России) является *RedShift* (www.redshift-live.com). Это действительно прекрасное приложение, которое понравится всем; а то, что на сайте имеется бесплатная (пусть и несколько урезанная) версия, только усиливает положительное впечатление. Однако нам не удалось найти хоть какие-нибудь сообщения об успешном запуске этого виртуального планетария в Linux. В каталоге приложений *Wine* (http://appdb.winehq.org) оно помечено как Garbage [Мусор], то есть никому пока не удалось получить от *Redshift* в Linux хоть какой-то положительный отклик. Следует отметить, что такая ситуация характерна для подавляющего большинства коммерческих приложений, связанных с различными областями науки.

> «Celestia — полноценная система моделирования Вселенной.»

Впрочем, в мире свободного ПО имеется продукт не хуже, а именно – *Celestia* (www.shatters.net/celestia). Это не просто кросс-платформенный настольный планетарий, а полноценная система знаний и моделирования Вселенной. Наличие возможности создания собственных скриптов, включающих текстуры и модели объектов, а также текст и звук, позволяет разрабатывать учебные материалы, которые будут не только познавательными, но и увлекательными. Поэтому мы предлагаем вам поближе познакомиться с *Celestia 1.6.0* и понять, как можно применять это приложение в качестве демонстрационного пособия на уроках астрономии или факультативных занятиях.

Установка

Celestia присутствует в репозиториях основных дистрибутивов Linux, так что откройте свой любимый менеджер пакетов, введите в строке поиска название приложения и установите его. Если же в вашем дистрибутиве его нет, можно скачать исходные тексты последней версии или двоичную сборку предыдущей с официального сайта. Компиляция из исходных текстов – не совсем удачная мысль, и связано это с тем, что современные дистрибутивы используют KDE 4 и *Qt4*, а *Celestia* создавалась в *Qt3* – таким образом, при сборке могут возникнуть проблемы.

Если вы все же отважились на компиляцию из исходных текстов, то она осуществляется стандартными командами, за исключением того, что при конфигурировании следует указывать используемую графическую среду: для KDE **./configure --with-kde**, для Gnome – **./configure --with-gnome**. В некоторых случаях для сборки в KDE следует сообщить каталог, где располагаются файлы библиотеки Qt: **./configure --with-kde --with-qt-includes=/usr/include/qt3**.

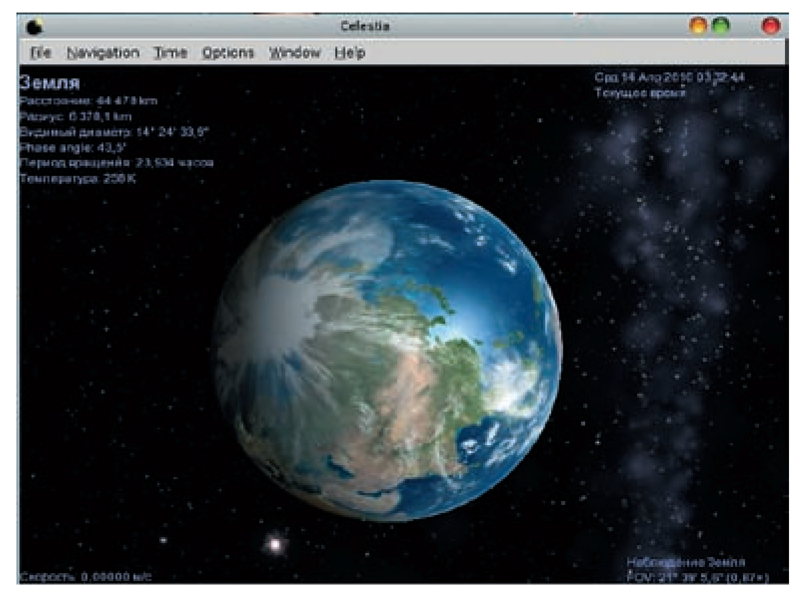
Итак, предположим, что приложение установлено и готово к работе. Первое, что мы увидим после запуска – это Солнце; затем камера перемещается к Земле, и можно начинать работу.

Для начала кратко пробежимся по пунктам меню.

» **File** [Файл] – здесь представлены команды, позволяющие открывать скрипты и URL, а также записывать на диск изображения и видеофрагменты.



 стр. 107



› После запуска Celestia перед вами предстает наша планета, в данном случае на фоне Млечного Пути.

» **Navigation** [Навигация] – основной пункт меню, в котором мы и будем проводить все время. Его команды позволяют искать и выбирать объекты, а также перемещаться к ним и устанавливать режимы наблюдения.

» **Time** [Время] – команды управления временем: установка, задание направления и скорости изменения.

» **Options** [Параметры] – здесь собраны команды, позволяющие настроить вид рабочего окна, а также включить/отключить вывод различных элементов: галактик, границ и названий созвездий, орбит и так далее.

» **Window** [Окно] – стандартные команды управления окном.

» **Help** [Справка] – информация о программе.

В последнем меню имеется пункт Run Demo [Запуск демо], выбор которого приводит к автоматическому открытию и выполнению скрипта, демонстрирующего основные возможности *Celestia*. В процессе работы скрипт можно приостановить (клавиша Пробел) или прервать (Esc). Можно просто смотреть сценарий и читать появляющийся текст, а можно активно вмешиваться в просмотр.

Автостопом по Вселенной

Управление осуществляется при помощи мыши и/или клавиатуры. Возможно также подключение джойстика.

» **Клавиатура:** клавиши Вправо-Влево вращают рабочее пространство в плоскости экрана, а клавиши Вверх-Вниз – в перпендикулярной плоскости, причем в качестве центра вращения выбран пользователь. Если выполнять те же действия при нажатой и удерживаемой клавише Shift, то в качестве центра вращения будет использоваться выделенный в данный момент объект.

» **Мышь:** удерживая левую кнопку мыши, можно перетаскивать точку обзора, а при нажатой правой кнопке мыши – вращать поле наблюдения. Если нажать и удерживать клавишу Shift, то перемещение мыши будет изменять масштаб отображения. Колесико мыши используется для перемещения вперед и назад.

Итак, для начала попробуем перемещаться по Вселенной самостоятельно. Например, отметим произвольную звезду или галактику левой кнопкой мыши, затем щелкнем по ней правой кнопкой, выберем Goto [Перейти] в появившемся меню или нажмем на клавиатуре клавишу G. Экран на какое-то время застынет, а затем начнется наше путешествие, и всего через несколько секунд выбранный объект во всем своем величии появится на экране, занимая большую его часть. Чтобы вернуться обратно к Солнцу, достаточно нажать клавиши H, а затем G, или выбрать пункты меню Navigation > Select Sol [Навигация > Выбрать Солнце] и Navigation > Go to Selection [Навигация > Перейти к выделенному].

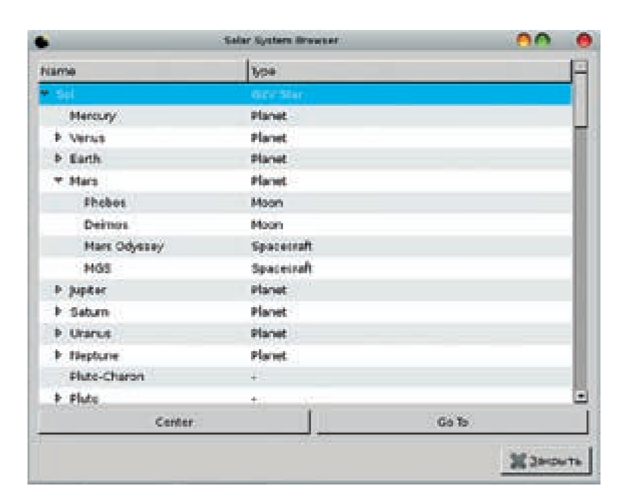
Для быстрого перемещения по Солнечной системе используются цифровые клавиши. Все планеты пронумерованы, то есть если нажать клавишу с цифрой 3, будет выбрана Земля, а с цифрой 8 – Нептун. Чтобы перейти к выбранному объекту, следует также нажать клавишу G. Кроме того, в *Celestia* встроен обозреватель звездных систем (Navigation > Solar System Browser). Иными словами, если выделен объект какой-либо звездной системы (чаще всего сама звезда), то в окне обозревателя отображаются все известные элементы, присутствующие в этой системе. Например, на рисунке ниже представлено состояние обозревателя для Солнечной системы. Видно, что около Марса вращаются не только естественные спутники (Moon), но и искусственные (Spacecraft). Если выбрать объект и нажать кнопку Goto, то можно увидеть его трехмерную модель. Кроме космических кораблей, планет и их спутников можно также посмотреть астероиды. Есть и обозреватель звезд (Navigation > Star Browser).

Если вы точно знаете, как называется интересующий вас объект, то можете осуществить его поиск. Для этого следует нажать клавишу Enter, после чего ввести название объекта (международное) в появившейся области внизу экрана. При вводе будет предлагаться до 12 вариантов названия, имеющихся в базе данных *Celestia*. Для перемещения по предлагаемым вариантам используется клавиша Tab. После ввода или выбора названия объекта нажмите клавишу Enter, а затем G, чтобы перейти к нему.

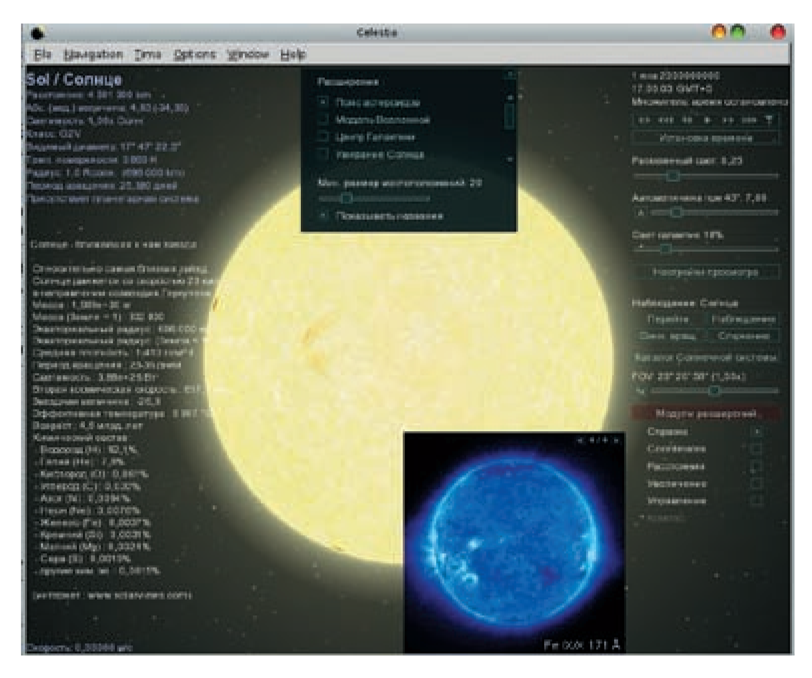
Celestia распознает названия планет, их спутников, а также наиболее известных звезд. Если необходимо найти галактику или обычную звезду, не имеющую собственного имени, то следует вводить обозначение по каталогу Гиппарха (HIP) или по каталогу NGC. Например, одной из самых интересных и ближайших к нам галактик является туманность Андромеды. Наиболее известно ее обозначение по каталогу Messier – M31: оно не распознается *Celestia*, однако ее можно найти по обозначению NGC 224. Нажмите клавишу Enter, введите **NGH 224**, затем нажмите Enter и следом клавишу G. Спустя несколько секунд перед вами откроется прекрасный вид на туманность Андромеды. И вновь, если захотите вернуться обратно в Солнечную систему, нажмите клавишу H, а затем G.

Туристические поездки

При помощи описанных средств можно изучать Вселенную очень долго, но всегда лучше иметь осведомленного гида, который проведет по самым интересным местам и расскажет занимательные истории. В качестве такового в *Celestia* используются скрипты (по поводу того, где их взять, смотрите врезку *Полезные ресурсы*). »



› Обозреватель звездной системы. Ух ты, у Марса уже четыре спутника! Два, правда, искусственных.



› **Дополнение LuaEDU в действии.**

Однако просто скачать скрипт мало: его еще необходимо правильно установить, поскольку многие сценарии содержат дополнительные текстуры и 3D-объекты.

Рассмотрим структуру каталогов Celestia. В нашем дистрибутиве Mandriva 2010.0 основной каталог располагается в **/usr/share/celestia**. Здесь лежат важные директории:

» **extras** – каталог, в котором чаще всего размещаются все дополнительные материалы: 3D-модели с техническим описанием, скрипты и сами пакеты расширений. Имеется также каталог **extras-standard**, в котором располагаются модели различных космических аппаратов.

» **models** – каталог для хранения 3D-моделей различных объектов.

» **scripts** – здесь, как правило, размещаются скрипты.

» **textures** – текстуры поверхностей планет и астероидов, структуры галактик и т.д. Внутри этой директории находятся три подкаталога: **hires** – для текстур высокого разрешения, **medres** – для текстур среднего разрешения (наиболее заполненный) и **lores** – для текстур низкого разрешения.

Кроме того, в **/usr/share/celestia** находятся следующие важные файлы:

» **celestia.cfg** – основной файл с настройками приложения. Здесь описаны пути ко всем файлам и каталогам, которые используются в работе Celestia. Если вы хотите подключить какой-либо каталог с дополнительными данными, то найдите здесь строку с ключом **ExtrasDirectories** и добавьте свой путь к уже имеющимся.

» **celestiaui.xml** – файл, описывающий состав главного текстового меню.

» **controls_ru.txt** – в этом файле определены все клавиатурные комбинации и их расшифровка на русском языке.

» **guide_ru.cel** – несмотря на такое же расширение, что и у скриптов, это просто текстовый файл, содержащий текст, который отображается в окне Тура по Солнечной системе (Navigation > Tour Guide)

» **demo_ru.cel** – демонстрационный скрипт, который выполняется при выборе пунктов меню Help > Run Demo или при нажатии клавиши **D**.

» **start_ru.cel** – скрипт, выполняющийся при каждом запуске *Celestia*.

Для организации занятия можно поступить следующим образом: заменить стартовый скрипт на свой собственный (просто переименуйте свой скрипт в **start_ru.cel**) или отредактировать в конфигурационном файле строку **InitScript**, заменив название стартового скрипта своим. В результате обучающиеся, придя на занятие, запустят приложение, и у них сразу же начнется демонстрация «правильного» сценария.

Как вы могли заметить, у многих файлов имеется суффикс **_ru**, показывающий, что это локализованный файл. В корневом каталоге также располагается и исходный файл без этого суффикса, который используется в ситуации, когда *Celestia* не может распознать язык базовой операционной системы.

Поскольку разработчики *Celestia*, по-видимому, больше ориентируются на платформу Windows, то в домашнем каталоге пользователя имеется только файл с настройками, и никаких подкаталогов не создается. Однако в файле **/usr/share/celestia.cfg** предписывается искать расширения и в **~/.celestia**. И все-таки дополнительные текстуры, скрипты и модели желательно размещать именно в описанных выше подкаталогах, то есть для расширения возможностей *Celestia* вам могут потребоваться права суперпользователя-root.

Разработчики скриптов также не оказывают особого внимания свободным системам, поэтому архив скрипта лучше всего для начала распаковать во временный каталог, а затем уже переписать полученные файлы в соответствующие каталоги установленной *Celestia* вручную. Мы подготовили для вас несколько архивов с «правильной» структурой подкаталогов. Для начала распакуйте файл **Tours.tar.gz**. В нем содержатся скрипты обзорных лекций по планетам солнечной системы, а также необходимые им файлы.

После распаковки в системном каталоге *Celestia* появляются скрипты с именами типа **SaturnTour.cel**. Давайте откроем один из них. Если *Celestia* не запущена, то вызовите ее, а затем выберите пункты меню File > Open Script [Файл > Открыть скрипт]. Перейдите в каталог **/usr/share/celestia** и выберите один из файлов, а затем нажмите кнопку Открыть. Скрипт начнет выполняться. Как указывалось ранее, для временной приостановки скрипта используется клавиша Пробел, а для его закрытия – клавиша Esc.

Скрипты – это простые текстовые файлы; вся выводимая пользователю информация содержится в команде **print**, поэтому их достаточно легко локализовать. В имеющемся наборе полностью переведен только тур по планете Марс.

Скрипты могут иметь два различных расширения: .cel и .celx. Первое используется для написания скриптов на соб-

Образовательный продукт для Windows

Информация на сайте программы и в сети Интернет наводит на мысль, что большинство пользователей и разработчиков *Celestia* предпочитают Windows. В результате многие скрипты и расширения полноценно работают только в этой операционной системе, а процедура их подключения в Linux довольно заковыриста.

Наиболее известным и полным образовательным расширением *Celestia* для Windows является *Celestia ED*, файлы которого мы пытаемся использовать в данной статье. Это даже не столько расширение, сколько самостоятельное приложение, созданное на базе *Celestia*. Размер установочного комплекта – 522 МБ, плюс набор дополнительных текстур высокого разрешения – 92 МБ. Загрузить их можно из каталога Motherlode (http://www.celestiamotherlode.net/catalog/educational.php).

В этом же каталоге присутствует 10 полноценных обучающих туров, включающих, помимо скриптов, текстовую и звуковую информацию, а также видеоролики в формате WMV. Полный объем информации превышает 3 ГБ, и потому на сайте можно купить DVD со всеми материалами.

К сожалению, эти ресурсы пока не русифицированы. Приглашаем энтузиастов взяться за работу.



› Кольцевая туманность M51. Кстати, вы знаете, что цвета на астрономических снимках — вещь условная?

ственном языке CEL, а второе – для сценариев, написанных на Lua (**LXF122—125**). В новых версиях скриптов и расширений все чаще применяют второй формат.

Именно на языке Lua написано одно из замечательных расширений – LUA EDU. Вы также можете найти его на диске – архив **LuaEdu_1.2.tar.gz**. Распакуйте его в корневой каталог **/usr/share/celestia**; если будет выдан запрос о том, заменить ли файлы, ответьте «Да». Для подключения этого пакета необходимо отредактировать конфигурационный файл *Celestia*: после слова **Configuration** и фигурной скобки, открывающей блок описания настроек, следует дописать

```
LuaHook "luahookinit.lua"
```

В архиве на диске находится уже измененный файл.

Запустите (или перезапустите) *Celestia*. У правого края экрана должна появиться новая панель (см. рисунок выше); если ее нет, то включите использование LuaEDU при помощи комбинации клавиш Shift+I. Иногда панель может не выводиться на экран только потому, что не умещается на нем: в этом случае просто увеличьте размер окна приложения.

На этой панели представлены наиболее часто используемые стандартные команды, а также добавлены новые. Например, на рисунке слева от Солнца находится достаточно подробная информация о нашем светиле. Это окно включается установкой крестика напротив пункта Справка на панели LuaEDU. Нажатие кнопки Модули расширений приводит к появлению окна, показанного на рисунке в центре вверху. В этом окне можно включить или отключить использование дополнительных инструментов, входящих в состав LuaEDU. Выберите Модель Вселенной, и вы будете ошеломлены масштабами нашего Мира. Кнопка Настройки просмотра позволит вам включить или отключить отображение различных классов объектов и их названий. Каталог объектов Солнечной системы позволит быстро выбрать желаемый и перейти к нему, нажав клавишу G. Установите крестик напротив пункта Управление, и вы сможете управлять видом, словно находитесь за штурвалом фантастического корабля.

Таким образом, LuaEDU представляет собой своеобразную панель управления планетарием, вынесенную на экран. Пакет достаточно удобен и функционален, а кроме того, добавляет в приложение дополнительную информацию и модели. Параметры LuaEDU можно изменить путем редактирования файла **config.lua**, расположенного в каталоге **extras/lua_edu_tools**. В этом же каталоге располагается файл **readme.txt**, поясняющий, как дополнить приложение своей собственной информацией и графическими материалами.

Полезные ресурсы

Если вас заинтересовал материал статьи и вы хотите расширить функциональность *Celestia* при помощи новых скриптов и дополнений, обратитесь к следующим сайтам:

- **The Celestia Motherlode** (www.celestiamotherlode.net) – самый крупный репозиторий дополнений для *Celestia*.
- **Celestia Matters** (www.celestialmatters.org) – ресурс высококачественных дополнений. Здесь даже имеется сценарий, показывающий единственный полет корабля Буран, а также космодром Байконур.
- **Celestia Texture Foundry** (www.shatters.net/~t00fri/texfoundry.php4) – хранилище текстур высокого разрешения.
- **Jack's Celestia Addons** (homepage.eircom.net/~jackcelestia) – небольшая коллекция дополнений.
- **Selden's List of Resources for Celestia** (www.lns.cornell.edu/~seb/celestia) – собрание всевозможных ссылок на ресурсы, посвященные *Celestia*: дополнения, текстуры, документацию и так далее. Некоторые из них, правда, уже устарели.

Кроме этого расширения, на языке Lua пишутся различные симуляторы – например, для 200-дюймового телескопа Хейла Паломарской обсерватории (http://www.lepp.cornell.edu/~seb/celestia/hale_telescope.html). Это дополнение предоставляет пользователю приборную доску управления телескопом и отображает на экране то, что он мог бы увидеть в обсерватории на самом деле.

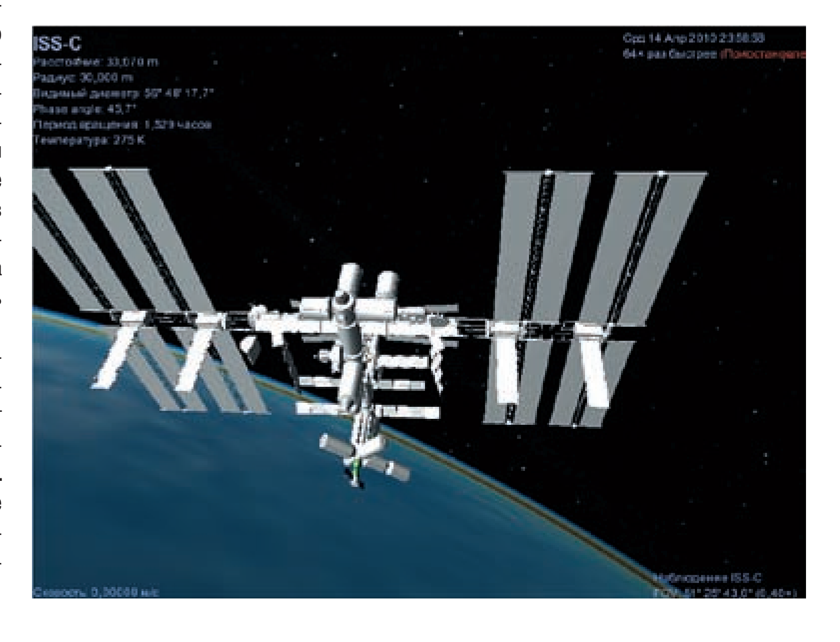
И вдобавок

Ну и в заключение позвольте показать вам два дополнения к *Celestia*, включенные в архив **Tours.tar.gz.** Если вы распаковали наш файл **Tours.tar.gz**, то выполните следующее: откройте *Celestia*, нажмите клавишу Enter и введите **Cygnus X1**. Нажмите Enter, а затем **G**, и через несколько секунд перед вами предстанет черная дыра из созвездия Лебедя. Вы увидите аккреционный диск и струи материи, бьющие вверх и вниз от диска.

А на десерт мы припасли нечто не совсем обычное. Нажмите клавишу Enter и введите Rigel Kentaurus A (достаточно набрать только Rigel, а затем выбрать соответствующий пункт при помощи клавиши Tab). Нажмите Enter, а затем клавишу G. Как только путешествие закончится и перед вами появится звезда, вновь нажмите Enter, а затем введите Pandora, нажмите Enter и затем клавишу G. По завершении перелета, перед вами предстанет мир «Аватара» Джеймса Кэмерона.

Загружайте дополнения, скрипты, модели и текстуры с различных сайтов и изучайте Вселенную. Поклонники фантастики смогут в полной мере оценить дополнения, позволяющие воссоздавать миры Звездных войн и Star Trek. Так обучение станет действительно увлекательным. LXF

› Оставьте проблемы Хьюстону: отслеживать местоположение Международной космической станции можно, не вставая с кресла.





В июньском номере

Меняем дистрибутивы

Мы покажем вам возможные альтернативы, а затем расскажем, как попробовать их все с минимальными неудобствами.

Межсетевые экраны

Для желающих достичь максимального уровня безопасности выделенный дистрибутив-брандмауэр просто необходим, но какой из них лучше?

Трюки с Google Calendar

У вас есть учетная запись Gmail? Тогда получайте на свой телефон бесплатные уведомления и много чего еще – мы подскажем, как это сделать.

Содержание последующих выпусков может меняться без уведомления.



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор
Валентин Синицын info@linuxformat.ru
Литературный редактор
Елена Толстякова
Переводчики
Илья Аввакумов, Александр Бикмеев, Юлия Дронова, Александр Казанцев, Ольга Кокорева, Светлана Кривошеина, Алексей Опарин, Валентин Развозжаев, Татьяна Цыганова
Редактор диска
Александр Кузьменков
Верстка, допечатная подготовка
Сергей Рогожников
Технический директор
Денис Филиппов
Генеральный директор
Павел Фролов
Учредители
Частные лица
Издатели
Виктор Федосеев, Павел Фролов

Отпечатано в типографии «Взлет»
196644, Санкт-Петербург, Колпинский район, пос. Саперный
Тел.: (812) 462-85-85
Заказ

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Пол Хадсон [Paul Hudson] paul.hudson@futurenet.com
Редактор обзоров Грэм Моррисон [Graham Morrison] graham.morrison@futurenet.com
Редактор диска Майк Сондерс [Mike Saunders] mike.saunders@futurenet.com
Художественный редактор Эфраин Эрнандес-Мендоса [Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com
Литературный редактор Эндрю Грегори [Andrew Gregory] agregory@futurenet.com
Подготовка материалов Марк Бэйн [Mark Bain], Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Энди Ченнел [Andy Channelle], Бен Мартин [Ben Martin], Боб Мосс [Bob Moss], Марк Нассел [Mark Nassal], Шашанк Шарма [Shashank Sharma], Адитья Шеваде [Aditya Shevade], Ник Вейч [Nick Veitch], Евгений Балдин, Александр Бикмеев, Антон Борисов, Андрей Боровский, Юрий Винник, Евгений Крестников, Владимир Лявшук, Сергей Петров, Дина Сенина, Алексей Федорчук, Роман Ярыженко
Художественные ассистенты Майк Бреннен [Mike Brennan], Карлтон Хибберт [Carlton Hibbert]
Иллюстрации Крис Хедли [Cris Hedley], Крис Винн [Chris WInn], iStockPhoto

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel +44 01225 442244 Email: linuxformat@futurenet.com
РОССИЯ:
Санкт-Петербург (редакция):
Лиговский пр., 50, корп. 15
Тел.: +7 (812) 309-06-86
Представительство в Москве:
Краснoказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)
Тел/факс: +7 (499) 271-49-54
По вопросам сотрудничества, партнерства, оптовых закупок:
partner@linuxcenter.ru